AÑO II - NUM. 31

MICROHOBBY SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO II - N.º 31

95 PTAS.



Canarias 105 ptas.

UTILIDADES

CARGADOR UNIVERSAL DE CODIGO MAQUINA



PROFANATION O LA MALDICION DE ABU SIMBEL

BASIC

GRAFICOS DEFINIDOS POR EL USUARIO

PROGRAMAS

AZATOTH

RANSOM

DESENSAMBLADOR

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Subdirector
Gabriel Nieto
Redactor Jefe
Africa Pérez Tolosa
Diseño
Rosa María Capitel
Redacción
José María Díaz, Miguel Angel Hijosa, Fco. Javier Martín
Secretaría Redacción
Carmen Santamaría
Colaboradores
Jesús Alonso, Lorenzo Cebeira, Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda
Fotografía
Javier Martínez, Carlos Candel
Portada
José María Ponce
Dibujos
Manuel Berrocal, J.R. Ballesteros, A. Perera, F.L. Frontán, J. Septien, Pejo, J.M. López Moreno

Edita
HOBBY PRESS, S.A.

Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión
Administrador General
Ernesto Marco

Jefe de Publicidad
Marisa Esteban
Secretaría de Publicidad
Concha Gutiérrez
Publicidad Barcelona
Isidro Iglesias
Tel.: (93) 307 11 13

Secretaría de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, n.º 8
Polígono Industrial de Alcobendas
Tel.: 654 32 11

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S.A. Valencia, 245
Barcelona

Imprime
Rotedic, S.A.
Carretera de Irún, Km. 12,450
Tel.: 734 15 00
Fotocomposición
Espacio y Punto, S.A.
Paseo de la Castellana, 268
Fotomecánica
Grof
Ezequiel Solana, 16
Depósito Legal:
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cia. Americana de Ediciones, S.R.L. Sud América, 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

MICROHOBBY no se hace necesariamente solidaria de las opiniones vertidas por sus colaboradores en los artículos firmados. Reservados todos los derechos.

Solicitado control OJD

MICROHOBBY ESTA SEMANA

AÑO II. N.º 31. 4 al 10 de junio de 1985
95 ptas. (Sobretasa Canarias 10 ptas.)

MICROHOBBY NUMEROS ATRASADOS

Queremos poner en conocimiento de nuestros lectores que para conseguir números atrasados de MICROHOBBY SEMANAL, no tienen más que escribirnos indicándonos en sus cartas el número deseado y la forma de pago elegida de entre las tres modalidades que explicamos a continuación.
Una vez tramitado esto, recibirá en su casa el número solicitado por el precio de 95 ptas., cada número, más 25 ptas. por gastos de envío.



FORMAS DE PAGO

- Enviando talón bancario nominativo a Hobby Press, S. A. al apartado de Correos 54062 de Madrid.
- Mediante Giro Postal, indicando número y fecha del mismo.
- Con Tarjeta de Crédito (VISA o MASTER CHARGE), haciendo constar su número y fecha de caducidad.

EL «AULA INFORMATICA» DE INVESTRONICA

Investrónica S. A. presentó recientemente lo que ellos denominan el «Aula Informática», basada en los microordenadores Sinclair (Spectrum y QL), que la compañía comercializa en España. Incorpora además, una serie de programas de desarrollo propio, tanto para tareas de gobierno de la Red Local, como para el soporte educacional propio de las tareas docentes que se realizarían en el Aula.

Para dicha aula se proporcionan todo tipo de materiales necesarios para su correcto funcionamiento, tanto a nivel de hardware como de software.

En este último apartado se incorpora un paquete que incluye programas de tres categorías:

a) Soporte de red y gobierno de las estaciones de trabajo desde el puesto central del profesor.

b) Software educacional. Incluye además los lenguajes LOGO, BASIC, y PASCAL, entre otros. También están disponibles aplicaciones de utilidad como puedan ser la creación de gráficos y el tratamiento de textos.



c) Software de desarrollo a base de unidades didácticas y programas de enseñanza.

Se ha elaborado una documentación completa sobre el modo de instalar este Aula Informática, que se incluye en la oferta que Investrónica realiza a los Centros Docentes.

QL & UNIX

Real Time Systems ha desarrollado un procedimiento que permite convertir programas de C y Pascal, para que puedan funcionar con los paquetes QDOS y CP/M-68 K.

El sistema sirve para la mayoría de los ordenadores 68000/68008 que incorporan el sistema operativo UNIX.

El paquete tiene dos compiladores C y Pascal, cada uno de los cuales incorpora una completa librería de utilidades para las máquinas QDOS.

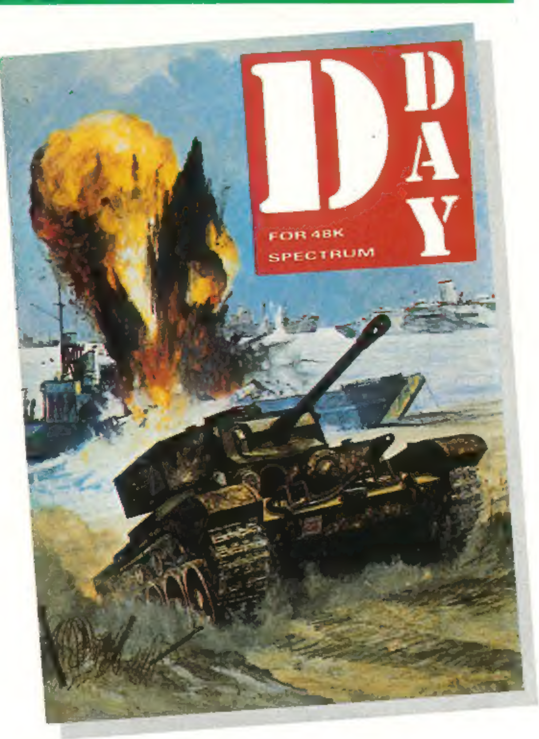
Quest Automation ha hecho la versión del CP/M-68 K que funciona con un paquete en disco para el QL de tal modo que los usuarios que lo posean, podrán correr sus programas utilizando el paquete RTS.

El «DAY D» PARA EL QL

La empresa de Software, Work Shop, ha decidido adentrarse en el mundo del QL, adaptando un conocido juego de estrategia de la compañía, concretamente el DAY D.

De este modo Work Shop pretende ser una de las primeras empresas de Software que realice un juego gráfico para el QL. El precio en Inglaterra será de aproximadamente 5.300 ptas., una cifra bastante alta, más aún si tenemos en cuenta que en nuestro país podría aparecer por casi el doble de esta cantidad.

El juego ya estaba disponible tanto para Spectrum, como para Commodore. Se trata de un «Wargame» que reproduce en el ordenador los acontecimientos del famoso Desembarco de Normandía, el golpe más duro asestado al ejército de Hitler durante la Segunda Gerra Mundial.



EN BUSCA DE PONCE

Muchas son las cartas que continuamente llegan a nuestra redacción felicitando a nuestro compañero Ponce por las portadas que, semanalmente, se publican en Microhobby, y no son menos los que confirman la gran expectación que surge con cada una de ellas para encontrar la habitualmente escondida firma del autor.

Este reto se ha complicado más en el número 30, en el que más que una firma, nuestro querido Ponce se ha marcado un jeroglífico.

Esperamos que esto sirva de mayor aliciente y que constituya un motivo más de adición a esta revista.



I FESTIVAL DE SOFTWARE

El centro Divulgador de la Informática de la Generalitat de Cataluña, ha organizado el primer Festival de Software, con un volumen de participación bastante alto. El plazo de entrega de programas acabó el pasado 30 de mayo, y el jurado, durante los próximos meses de junio y julio, hará la preselección de todos los programas que participan. Aquellos que finalmente sean seleccionados, serán presentados al público en el Salón de la Imagen, el Sonido y la Electrónica (SONIMAG) que se celebrará en los primeros días de octubre del presente año.

Los premios serán de 100.000 ptas. para los primeros de cada categoría y en las dos modalidades, junior o senior.

El público en general tendrá la posibilidad de votar a los mejores programas, a través de unos impresos que repartirá en su momento la organización.

Los tipos de programas que se han presentado han sido juegos, de aprendizaje o iniciación, enseñanza asistida por ordenador, vida práctica, arte y cultura, y simulación.

Una buena iniciativa.



LIBROS



FICHEROS EN BASIC

Paraninfo. C. Delannoy. 173 páginas.

Hay muchos usuarios de ordenador que lo primero que se plantean a la hora de utilizarlo es el manejo de ficheros, por muchas razones que van desde una necesidad hasta un claro deseo de encauzar sus conocimientos por el terreno de la gestión.

El libro de C. Delannoy nos proporciona los medios necesarios para la creación y manejo de ficheros, explicándonos, además, la diferencia que existe entre un fichero manual y uno informático.

Tras la consabida introducción que no puede faltar en ninguna publicación que se considire medianamente seria, y que en este caso pasa revista a lo que es un fichero informático, y dentro de éste el acceso secuencial, directo e indexado, comenzamos a dar nuestros primeros pasos en la creación de un fichero. Y es precisamente esto, lo que se nos va a tratar de explicar durante toda la obra.

En el capítulo de los Ficheros Secuenciales se explica, por ejemplo, como se crea un fichero directorio, al principio con un programa simplificado bastante sencillo y posteriormente, con otro más general que consiste en un repertorio telefónico.

En lo dos capítulos siguientes se amplian aún más los conceptos del fichero secuencial con el fin de obtener así un repertorio más completo: grabación de información, la lectura y los modos separadores, el IMPUT LINE, la utilización del PRINT en relación al fichero y el listado de un fichero repertorio. Todo ello acompañado de un gran variedad de ejemplos.

El capítulo 5, trata del concepto MENU y de cómo debemos construir el nuestro de una forma que resulte sencilla y sobre todo, práctica a la hora de crear.

También se explican, en sucesivos capítulos, qué son los ficheros de acceso directo y el modo de crearlos, explicando además las ventajas y desventajas con respecto al secuencial.

El último fichero que trata este libro, es el Indexado, en el que se explica en qué consisten los índices de memoria.

Se trata de una obra indispensable para todos aquellos que quieren aprender el modo de crear sus propios ficheros. En el caso de los usuarios de Spectrum, estos deberán de adecuar al Basic de este ordenador algunos de los programas demostrativos que hay en el libro, que han sido escritos en un BASIC más generalizado.

LETRA NEGRITA

El juego de caracteres del Spectrum fue concebido fundamentalmente de cara a la utilidad, es decir, tal vez no se prestó demasiada atención al aspecto de las letras. Para mantener la legibilidad de los caracteres y aumentar su estética, nuestro amigo **Benjamín Amador** nos manda una rutina desde Jaca (Huesca), escrita en lenguaje máquina, que nos permitirá escribir con letra negrita.

Como siempre, damos el listado en lenguaje ensamblador para los interesados en profundizar en el algoritmo elegido y un pequeño cargador Basic para los que, simplemente, quieren que funcione ya. Como se puede observar, la forma de hacerlo es bastante simple; en el listado en ensamblador, la clave se encuentra en las instrucciones SRA A y OR (HL); la primera desplaza a la derecha el byte que estamos considerando y la segunda, al efectuar la operación OR, lo que realiza es una suma del byte original con el que anteriormente hemos desplazado, consiguiendo como resultado final, una letra más gruesa que la original.



A. PERERA

Naturalmente, el nuevo juego de caracteres así creado se almacena en memoria RAM tras cambiar el puntero de la variable CHARS.

Para volver a la situación original, es preciso efectuar dos POKES:

POKE 23606,0 y POKE 23607,60

esto nos devolverá al juego de caracteres propio del Spectrum.

```
 10          ORG   65338
 20 ;
 30          LD    HL,#3D00
 40 ;
 50 ;FICHERO DE FORMA DE
 60 ;CARACTERES(#3D00-#3FFF)
 70 ;
 80          LD    DE,#FC39
 90 ;
100 ;NUEVA DIRECCION DE
110 ;CARACTERES EN RAM
120 ;
130          LD    BC,#0300
140 ;
150 ;NUMERO DE BYTES DEL
160 ;FICHERO DE CARACTERES
170 ;
180 ;COMIENZO DE LA RUTIN/
190 ;QUE ELONGA EL CARACT
200 ;
210 SIG      LD    A,(HL)
220          SRA   A
230          OR    (HL)
240          LD    (DE),A
250          INC   DE
260          INC   HL
270          DEC   BC
280          LD    A,C
290          OR    B
300          JR    NZ,SIG
310 ;
320 ;FIN DE LA RUTINA DE
330 ;ELONGACION
340 ;
350          LD    HL,#FB39
360          LD    (#5C36)
370 ;
380 ;#5C36 VARIABLE DEL
390 ;SISTEMA CHARS
400 ;
410          RET
```

```
9910 CLEAR 64568: FOR f=65338 TO
 65365: READ a: POKE f,a: NEXT f
9920 RANDOMIZE USR 65338
9930 DATA 33,0,61,17,57,252,1,0,
3,126,203,47,182,18,19,35,11,121
,176,32,244,33,57,251,34,54,92,2
01
```

AZATOTH

José Luis VILLALBA

Spectrum 16 K

El escenario, un inmenso y tenebroso castillo plagado de trampas, guardianes, puertas que se cierran y un sin fin de objetos que tendremos que ir recogiendo hasta encontar nuestro objetivo.

Y este no es otro que la copa encantada con la que podremos salir del endiablado castillo y evitar que el malvado mago que habita en él, nos deje encerrados en sus innumerables habitaciones con su maléfica influencia.

Para evitarlo, tendremos que ir recorriendo todas las habitaciones y recoger en ellas cuantos objetos nos encontremos evitando, también, que se nos cierren todas las salidas y quedarnos atrapados con lo que perderíamos una vida (disponemos de cinco para llegar a la copa).

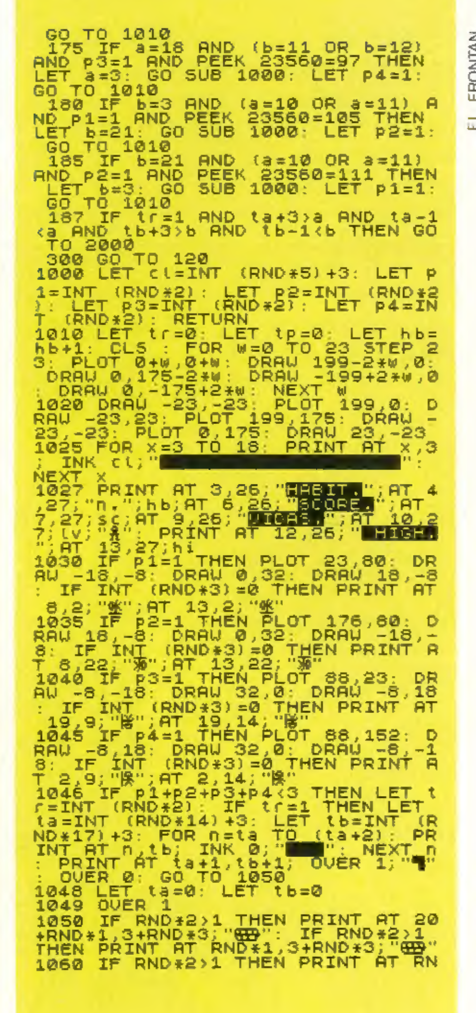
Podemos elegir el nivel de dificultad al inicio del juego, de lo que dependerá que la copa se encuentre en una habitación más proxima o más lejana.

Mucha suerte, la vas a necesitar.

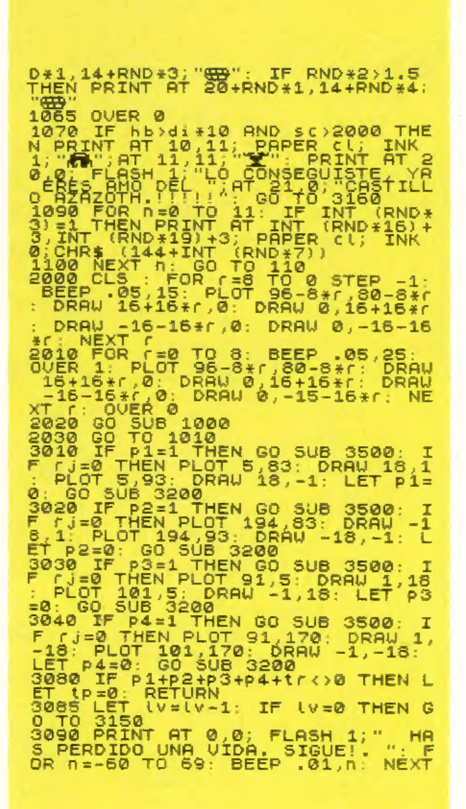
NOTAS GRAFICAS

A B C D E F G H I J K L M N O P
Q R S T U

```
  50 LET hi=0: GO SUB 5000
  70 PRINT "DEBES ENCONTRAR ESTA
 COPA    ": PRINT "QUE ESTA P
ERDIDA .          ": PRINT '
"CUIDADO CON QUEDARTE ENCERRADO.
 ": PRINT ''"Q-ARRIBA","A-ABAJO"
,"O-DERECHA","I-IZQUIERDA": PRIN
T '''"PULSA ENTER": PAUSE 0
  80 BORDER 0: PAPER 0: INK 7: C
LS
  85 INPUT "NIVEL DE DIFICULTAD
(1-10)? ";di: IF di<1 OR di>10 T
HEN GO TO 85
  90 LET cl=INT (RND*5)+3: LET s
c=0
 100 LET lv=5: LET tr=0: LET a=1
0: LET b=13: LET hb=0: LET p1=1:
 LET p2=1: LET p3=1: LET p4=1
 105 GO TO 1010
 110 PRINT AT a,b; INK cl; OVER
1;"": OVER 0
 120 LET tp=tp+1: LET pk=PEEK 23
560
 125 IF pk=97 AND a<18 THEN PRIN
T AT a,b; INK cl;"■": LET a=a+1:
 PRINT AT a,b; INK cl; OVER 1;"
"
 130 IF pk=113 AND a>3 THEN PRIN
T AT a,b; INK cl;"■": LET a=a-1:
 PRINT AT a,b; INK cl; OVER 1;"
"
 140 IF pk=105 AND b>3 THEN PRIN
T AT a,b; INK cl;"■": LET b=b-1:
 PRINT AT a,b; INK cl; OVER 1;"
"
 150 IF pk=111 AND b<21 THEN PRI
NT AT a,b; INK cl;"■": LET b=b+1
: PRINT AT a,b; INK cl; OVER 1;"
"
 155 IF POINT (b*8+4,(21-a)*8+2)
=0 THEN BEEP .005,55: PRINT AT a
,b;"■";AT a,b; OVER 1;"": LET s
c=sc+17: PRINT AT 7,27;sc
 165 BEEP .0009,16: IF tp>19 AND
 INT (RND*3)=0 THEN GO SUB 3000
 170 IF a=3 AND (b=11 OR b=12) A
ND p4=1 AND PEEK 23560=113 THEN
LET a=18: GO SUB 1000: LET p3=1:
```



```
 GO TO 1010
 175 IF a=18 AND (b=11 OR b=12)
AND p3=1 AND PEEK 23560=97 THEN
LET a=3: GO SUB 1000: LET p4=1:
GO TO 1010
 180 IF b=3 AND (a=10 OR a=11) A
ND p1=1 AND PEEK 23560=105 THEN
LET b=21: GO SUB 1000: LET p2=1:
 GO TO 1010
 185 IF b=21 AND (a=10 OR a=11)
AND p2=1 AND PEEK 23560=111 THEN
 LET b=3: GO SUB 1000: LET p1=1:
 GO TO 1010
 187 IF tr=1 AND ta+3>a AND ta-1
<a AND tb+3>b AND tb-1<b THEN GO
 TO 2000
 300 GO TO 120
1000 LET cl=INT (RND*5)+3: LET p
1=INT (RND*2): LET p2=INT (RND*2
): LET p3=INT (RND*2): LET p4=IN
T (RND*2): RETURN
1010 LET tr=0: LET tp=0: LET hb=
hb+1: CLS : FOR w=0 TO 23 STEP 2
3: PLOT 0+w,0+w: DRAW 199-2*w,0:
 DRAW 0,175-2*w: DRAW -199+2*w,0
: DRAW 0,-175+2*w: NEXT w
1020 DRAW -23,-23: PLOT 199,0: D
RAW -23,23: PLOT 199,175: DRAW -
23,-23: PLOT 0,175: DRAW 23,-23
1025 FOR x=3 TO 18: PRINT AT x,3
; INK cl;"■■■■■■■■■■■■■■■■■■■":
NEXT x
1027 PRINT AT 3,26;"HABIT.";AT 4
,27;"n.";hb;AT 6,26;"SCORE.";AT
7,27;sc;AT 9,26;"VIDAS.";AT 10,2
7;lv;"": PRINT AT 12,26;" HIGH.
";AT 13,27;hi
1030 IF p1=1 THEN PLOT 23,80: DR
AW -18,-8: DRAW 0,32: DRAW 18,-8
: IF INT (RND*3)=0 THEN PRINT AT
 8,2;"";AT 13,2;""
1035 IF p2=1 THEN PLOT 176,80: D
RAW 18,-8: DRAW 0,32: DRAW -18,-
8: IF INT (RND*3)=0 THEN PRINT A
T 8,22;"";AT 13,22;""
1040 IF p3=1 THEN PLOT 88,23: DR
AW -8,-18: DRAW 32,0: DRAW -8,18
: IF INT (RND*3)=0 THEN PRINT AT
 19,9;"";AT 19,14;""
1045 IF p4=1 THEN PLOT 88,152: D
RAW -8,18: DRAW 32,0: DRAW -8,-1
8: IF INT (RND*3)=0 THEN PRINT A
T 2,9;"";AT 2,14;""
1046 IF p1+p2+p3+p4<3 THEN LET t
r=INT (RND*2): IF tr=1 THEN LET
ta=INT (RND*14)+3: LET tb=INT (R
ND*17)+3: FOR n=ta TO (ta+2): PR
INT AT n,tb; INK 0;"■■■": NEXT n
: PRINT AT ta+1,tb+1; OVER 1;"■"
: OVER 0: GO TO 1050
1048 LET ta=0: LET tb=0
1049 OVER 1
1050 IF RND*2>1 THEN PRINT AT 20
+RND*1,3+RND*3;"": IF RND*2>1
THEN PRINT AT RND*1,3+RND*3;""
1060 IF RND*2>1 THEN PRINT AT RN
```



```
D*1,14+RND*3;"": IF RND*2>1.5
THEN PRINT AT 20+RND*1,14+RND*4;
""
1065 OVER 0
1070 IF hb>di*10 AND sc>2000 THE
N PRINT AT 10,11; PAPER cl; INK
1;"";AT 11,11;"": PRINT AT 2
0,0; FLASH 1;"LO CONSEGUISTE, YA
 ERES AMO DEL ";AT 21,0;"CASTILL
O AZAZOTH.!!!!!": GO TO 3160
1090 FOR n=0 TO 11: IF INT (RND*
3)=1 THEN PRINT AT INT (RND*16)+
3,INT (RND*19)+3; PAPER cl; INK
0;CHR$ (144+INT (RND*7))
1100 NEXT n: GO TO 110
2000 CLS : FOR r=8 TO 0 STEP -1:
 BEEP .05,15: PLOT 96-8*r,80-8*r
: DRAW 16+16*r,0: DRAW 0,16+16*r
: DRAW -16-16*r,0: DRAW 0,-16-16
*r: NEXT r
2010 FOR r=0 TO 8: BEEP .05,25:
OVER 1: PLOT 96-8*r,80-8*r: DRAW
 16+16*r,0: DRAW 0,16+16*r: DRAW
 -16-16*r,0: DRAW 0,-15-16*r: NE
XT r: OVER 0
2020 GO SUB 1000
2030 GO TO 1010
3010 IF p1=1 THEN GO SUB 3500: I
F rj=0 THEN PLOT 5,83: DRAW 18,1
: PLOT 5,93: DRAW 18,-1: LET p1=
0: GO SUB 3200
3020 IF p2=1 THEN GO SUB 3500: I
F rj=0 THEN PLOT 194,83: DRAW -1
8,1: PLOT 194,93: DRAW -18,-1: L
ET p2=0: GO SUB 3200
3030 IF p3=1 THEN GO SUB 3500: I
F rj=0 THEN PLOT 91,5: DRAW 1,18
: PLOT 101,5: DRAW -1,18: LET p3
=0: GO SUB 3200
3040 IF p4=1 THEN GO SUB 3500: I
F rj=0 THEN PLOT 91,170: DRAW 1,
-18: PLOT 101,170: DRAW -1,-18:
LET p4=0: GO SUB 3200
3080 IF p1+p2+p3+p4+tr<>0 THEN L
ET tp=0: RETURN
3085 LET lv=lv-1: IF lv=0 THEN G
O TO 3150
3090 PRINT AT 0,0; FLASH 1;"  HA
S PERDIDO UNA VIDA. SIGUE!. ": F
OR n=-60 TO 69: BEEP .01,n: NEXT
```



FL FRONTAN

```
 n: GO SUB 1000: GO TO 1010
3100 LET tp=0: RETURN
3150 PRINT AT 0,0; BRIGHT 1;"YA
NO SALDRAS NUNCA.";AT 10,27;"0"
3160 IF sc>hi THEN LET hi=sc: PR
INT AT 13,27;hi
3170 PRINT ; FLASH 1;#0;" PULSA
UNA TECLA PARA EMPEZAR.  ": FOR
n=0 TO 60: BEEP .009,n: NEXT n:
PAUSE 0: PAUSE 0: GO TO 80
3200 FOR n=1 TO 4: BEEP .005,-5*
n: NEXT n: RETURN
3500 LET rj=INT (RND*3): RETURN
3510 RETURN
5000 FOR n=0 TO 20: READ w$
5010 FOR m=0 TO 7
5020 READ a: POKE USR w$+m,a
5030 NEXT m: NEXT n
5040 DATA "r",24,44,44,24,254,36
,36,108,"s",24,52,52,24,127,36,3
6,54,"t",24,60,60,24,126,36,36,3
6,"u",24,36,60,24,126,36,36,102
5050 DATA "k",0,127,145,145,255,
136,72,63,"l",0,252,18,18,255,13
4,132,252,"g",8,24,254,136,156,5
4,50,60
5060 DATA "a",255,253,122,122,52
,16,60,126,"b",110,145,44,86,170
,84,40,16,"c",11,12,8,28,54,54,5
8,28,"m",156,162,190,136,254,137
,148,162
5070 DATA "e",56,16,16,56,68,68,
124,124,"f",8,8,20,36,66,231,239
,70,"d",2,3,4,4,60,246,255,62
5075 DATA "h",162,148,137,254,13
6,190,162,156,"i",255,16,150,65,
61,85,150,32,"j",4,105,170,168,1
70,105,8,255
5080 DATA "n",15,56,63,63,252,18
4,154,158,"o",240,12,252,252,127
,125,121,121,"p",158,248,15,15,3
,1,7,63,"q",121,63,240,240,192,1
28,224,252
5500 RETURN
```

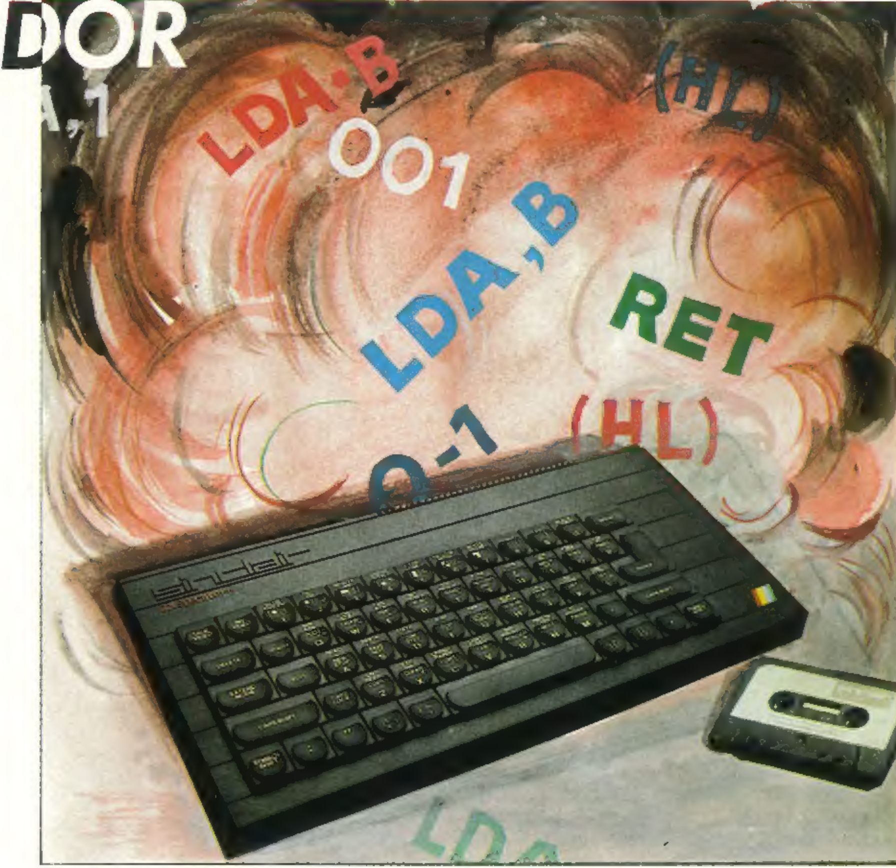
DESENSAMBLADOR

Antonio OCON CARRERAS

Spectrum 48 K

Como su propio nombre indica, se trata de un desensamblador de código máquina con la peculiaridad de ofrecer todos los números en notación decimal y no hexadecimal, como la mayoría de los desensambladores existentes en el mercado.

Para su funcionamiento, nos pedirá una dirección de comienzo de la rutina a desensamblar y la longitud de ésta en bytes, y a continuación, aparecerá el listado de los nemónicos o memóricos



de lenguaje ensamblador correspondiente a dicha rutina, como viene indicado en las instrucciones al final del programa.

El desensablador puede interrumpirse por dos razones:

— La zona a desensamblar no contiene una rutina en código máquina, sino un programa basic, una pila de datos, etc...

— Existe una rutina en esta zona, pero se ha situado al comienzo de la lectura en medio de una instrucción de más de un byte, por lo que el desensamblado no es correcto, pudiéndose llegar, de esta manera, a una instrucción inexistente.

```
   1 REM ************************
   2 REM   DESENSAMBLADOR Z-80
   3 REM   © ANTONIO OCON-1985
   4 REM ************************

   9 GO TO 9900
  10 CLS
  19 REM ** ENTRADA DE DATOS **
  20 PRINT "     INTRODUCCION DE
DATOS": INPUT "Direccion de comi
enzo..";d: LET n=d
  30 INPUT "Numero de Bytes..";f
: LET f=f+d
  35 LET s=0: LET p=2
  40 CLS : PRINT "DIRECCION  INS
TRUCCION
_______________________________"
 100 REM ** LECTURA **
 110 LET a=PEEK n
 120 LET n2=0: LET inded=0: LET
indix=0: LET n$=""
 125 REM ** es IX o IY? **
 127 IF a=221 THEN GO TO 3010
 128 IF a=253 THEN GO TO 3020
 130 REM ** Es CB? **
 140 IF a=203 THEN GO TO 1000
 150 REM ** Es ED? **
 160 IF a=237 THEN GO TO 2000
 170 REM ** ES ITERATIVA? **
 180 IF a>63 AND a<192 THEN GO T
O 400
 190 REM ** CASO NORMAL **
 200 LET m=INT (a/10)
 210 LET r=a-10*m
 220 RESTORE (9000+m+inded)
 230 FOR i=0 TO r
 240 READ a$
 250 NEXT i
 260 REM ** LLEVA NUMERO? **
 270 IF a$(1)="2" THEN GO SUB 75
0
 280 IF a$(1)="1" THEN GO SUB 70
0
 290 REM ** IMPRESION **
 295 IF indix<>0 THEN GO TO 1270
 300 IF A$(1)="*" THEN LET A$=A$
(2 TO )+")"
 301 PRINT TAB 3;N;TAB 12;a$
 303 LET p=p+1: IF p=22 THEN LET
 p=0: PRINT #1;"          COPY (S
/N)": LET dirsi=306: LET dirno=3
08: GO TO 9500
 304 GO TO 308
 306 COPY : CLS
 308 IF indix<>0 THEN LET indix=
indix-1
 310 LET n=n+n2+1+indix
 320 IF n>f THEN GO TO 340
 330 GO TO 100: REM goto Lectura

 339 REM **      FIN        **
 340 PRINT #1;"   Otra Direccion
 (S/N)?": LET dirsi=10: LET dirn
o=350: GO TO 9500
 350 PRINT "   Teclee RUN para c
ontinuar": STOP
 399 REM ** CALCULOS ITERATIVOS*
*
 400 REM ** a>63 y a<192 **
 410 IF a>127 THEN GO TO 540
 420 REM ** a>63 y a<128 **
 430 LET m=INT ((a-64)/8)
 440 LET r=a-64-(m*8)
 450 RESTORE 9110
 460 FOR i=0 TO m
 470 READ a$: NEXT i
 475 RESTORE 9110
 480 FOR i=0 TO r
 490 READ b$: NEXT i
 500 LET a$="ld "+a$+","+b$
 510 IF a=118 THEN LET a$="halt"
 520 GO TO 290: REM goto imprime
 540 REM ** a>127 **
 550 LET m=INT ((a-128)/8)
 560 LET r=a-128-(m*8)
 570 RESTORE 9100
 580 FOR i=0 TO m
 590 READ a$: NEXT i
 600 RESTORE 9110
 610 FOR i=0 TO r
 620 READ b$: NEXT i
 630 LET a$=a$+b$
 640 GO TO 290: REM goto imprime

 700 REM ** SUB. ANADE 1 NO. **

 710 LET n2=1
 720 LET num=PEEK (n+1)
 730 IF a=211 THEN LET a$="out (
"+STR$ (num)+"),a": RETURN
 740 GO TO 780

 750 REM ** SUB. ANADE 2 NOS. **

 760 LET n2=2
 770 LET num=PEEK (n+2)*256+PEEK
 (n+1)
 780 LET a$=a$(2 TO )+STR$ (num)
 790 IF a=34 THEN LET a$="ld ("+
STR$ (num)+"),hl"
 800 IF a=50 THEN LET a$="ld ("+
STR$ (num)+"),a"
 810 IF inded=0 THEN RETURN
 820 IF a=67 THEN LET a$="ld ("+
STR$ (num)+"),bc": RETURN
 825 IF a=83 THEN LET a$="ld ("+
STR$ (num)+"),de": RETURN
 830 IF a=99 THEN LET a$="ld ("+
STR$ (num)+"),hl": RETURN
 840 IF a=115 THEN LET a$="ld ("
+STR$ (num)+"),sp": RETURN
 850 RETURN
1000 REM **         CB          **
1020 LET n=n+1
1030 LET a=PEEK n
1040 LET m=INT (a/8)
1050 LET r=a-(m*8)
1060 IF a>63 THEN GO TO 1160
1070 REM ** CB<64 **
1080 RESTORE 9120
1090 FOR i=0 TO m
1100 READ a$: NEXT i
1110 RESTORE 9110
1120 FOR i=0 TO r
1130 READ b$: NEXT i
1140 LET a$=a$+b$
1150 GO TO 1260: REM goto imprim
e
1160 REM ** CB>64 **
1170 LET y=INT (a/64)-1
1180 LET x=m-(8*(y+1))
1190 RESTORE 9130
1200 FOR i=0 TO y
1210 READ a$: NEXT i
1220 RESTORE 9110
1230 FOR i=0 TO r
1240 READ b$: NEXT i
1250 LET a$=a$+STR$ (x)+","+b$
1260 IF indix=0 THEN GO TO 290:
REM goto imprime

1270 REM ** CAMBIA HL-(IX+D) **
1280 LET top=LEN a$-1
1290 FOR c=1 TO top
1300 LET p$=a$(c TO c+1)
1310 IF p$="hl" THEN GO TO 1340
1320 NEXT c
1325 REM **      ERROR      **
1330 BEEP .5,-15: BEEP .5,-20: P
RINT : PRINT BRIGHT 1;"Desensamb
lado imposible en esta  zona,rev
ise la rutina e intro- duzca otr
a direccion": STOP
1340 LET a$=a$(1 TO c-1)+i$+n$+a
$(c+2 TO )
1350 GO TO 300: REM goto imprime

2000 REM  **        ED         **

2020 LET n=n+1
2030 LET a=PEEK n
2035 IF a<64 THEN GO TO 2120: RE
M      goto error
2040 IF a<124 THEN LET inded=200
:               GO TO 190: REM
   goto caso normal
2100 REM ** ED>160 **
2110 IF a<160 THEN GO TO 1330: R
EM goto error
2120 LET inded=300
2130 GO TO 190: REM caso normal

3000 REM **       IX o IY      **

3010 LET i$="IX": GO TO 3030
3020 LET i$="IY"
3030 LET n=n+1
3040 LET a=PEEK (n)
3045 REM * es IX-caso especial?*
3050 IF a=54 THEN LET a$="ld ("+
i$+"+"+STR$ (PEEK (n+1))+"),"+ST
R$ (PEEK (n+2)): LET indix=3: GO
 TO 300
3055 IF a=41 THEN LET a$="add "+
i$+","+i$: LET indix=1: GO TO 30
0
3060 REM ** es IX-CB ? **
3070 IF a=203 THEN LET n$="+"+ST
R$ (PEEK (n+1)): LET a=PEEK (n+2
): LET indix=3: GO TO 1040
3080 REM ** es IX-d ? **
3085 IF a=57 THEN GO TO 3100
3090 IF a>43 AND a<191 THEN LET
n$="+"+STR$ (PEEK (n+1)): LET in
dix=2: GO TO 180
3100 REM ** IX-caso normal **
3200 LET indix=1: GO TO 180

8999 REM **datas normales**

9000 DATA "nop","2ld bc,","ld (b
c),a","inc bc","inc b","dec b","
1ld b,","rlca","ex af,af'","add
hl,bc"
9001 DATA "ld a,(bc)","dec bc","
inc c","dec c","1ld c,","rrca","
1djnz DIS-","2ld de,","ld (de),a
","inc de"
9002 DATA "inc d","dec d","1ld d
,","rla","1jr DIS-","add hl,de",
"ld a,(de)","dec de","inc e","de
c e"
9003 DATA "1ld e,","rra","1jr nz
,DIS-","2ld hl,","2","inc hl","i
nc h","dec h","1ld h,","daa"
9004 DATA "1jr z,DIS-","add hl,h
l","2*ld hl,(","dec hl","inc l",
"dec l","1ld l,","cp l","1jr nc,
DIS-","2ld sp,"
9005 DATA "2","inc sp","inc (hl)
","dec (hl)","1ld (hl),","scf","
1jr c,DIS-","add hl,sp","2*ld a,
(","dec sp"
9006 DATA "inc a","dec a","1ld a
,","ccf"
9019 DATA " "," ","ret nz","pop
bc","2jp nz,","2jp ","2call nz,"
,"push bc","1add a,","rst 0"
9020 DATA "ret z","ret","2jp z,"
," ","2call z,","2call ","1adc a
,","rst 8","ret nc","pop de"
9021 DATA "2jp nc,","1","2call n
c,","push de","1sub ","rst '16'"
,"ret c","exx","2jp c,","1*in a,
("
9022 DATA "2call c,"," ","1sbc a
,","rst '24'","ret po","pop hl",
"2jp po,","ex (sp),hl","2call po
,","push hl"
9023 DATA "1and ","rst '32'","re
t pe","jp (hl)","2jp pe,","ex de
,hl","2call pe,"," ","1xor ","rs
t '40'"
9024 DATA "ret p","pop af","2jp
p,","di","2call p,","push af","1
or ","rst '48'","ret m","ld sp,h
l"
9025 DATA "2jp m,","ei","2call m
,"," ","1cp ","rst '56'"

9099 REM **datas iterativas**
9100 DATA "add a,","adc a,","sub
 ","sbc a,","and ","xor ","or ",
"cp "
9110 DATA "b","c","d","e","h","l
","(hl)","a"
9120 DATA "rlc ","rrc ","rl ","r
r ","sla ","sra "," ","srl "
9130 DATA "bit ","res ","set "

9205 REM **datas de ED<124**

9206 DATA " "," "," "," ","in b,
(c)","out (c),b","sbc hl,bc","2"
,"neg","retn"
9207 DATA "im 0","ld i,a","in c,
(c)","out (c),c","adc hl,bc","2l
d bc,("," ","reti"," ","ld r,a"
9208 DATA "in d,(c)","out (c),d"
,"sbc hl,de","2"," "," ","im 1",
"ld a,i","in e,(c)","out (c),e"
9209 DATA "adc hl,de","2ld de,("
," "," ","im 2","ld a,r","in h,(
c)","out (c),h","sbc hl,hl","2"
9210 DATA " "," "," ","rrd","in
l,(c)","out (c),l","adc hl,hl","
2*ld hl,("," "," "
9211 DATA " ","rld","in f,(c)","
","sbc hl,sp","2"," "," "," ","
"
9212 DATA "in a,(c)","out (c),a"
,"adc hl,sp","2*ld sp,("

9300 REM  **datas de ED>124**

9316 DATA "ldi","cpi","ini","out
i"," "," "," "," ","ldd","cpd"
9317 DATA "ind","outd"," "," ","
"," ","ldir","cpir","inir","oti
r"
9318 DATA " "," "," "," ","lddr"
,"cpdr","indr","otdr"," "," "

9500 REM **   RUTINA (S/N)     **
9510 IF INKEY$="" THEN GO TO 951
0
9520 LET c$=INKEY$
9525 BEEP .2,40
9530 IF c$="S" OR c$="s" THEN GO
 TO dirsi
9540 CLS : GO TO dirno

9900 REM **   PRESENTACION     **

9910 CLS : PRINT "     DESENSAMBL
ADOR DECIMAL            © Antonio
 Ocon - 1985       Este programa
ofrece el listado de los mnemoni
cos correspondien-tes  al codigo
 de maquina junto  con la direcc
ion de la instruc- cion principa
l."
9911 PRINT : PRINT "Si se detien
e el listado con un error es deb
ido a que se intenta desensambla
r una seccion de la memoria que
no contiene un pro- grama en cod
igo de maquina,como por ejemplo,
tablas de datos,zonade programas
 BASIC,etc."
9912 PRINT : PRINT "Todos los nu
meros representados son decimale
s,lo que facilita lalocalizacion
 de rutinas,asi comola identific
acion de valores."
9913 PRINT : PRINT BRIGHT 1;"
     PULSE UNA TECLA            "
9914 IF INKEY$="" THEN GO TO 991
4
9915 BEEP .2,40: GO TO 10
```

El secreto del templo

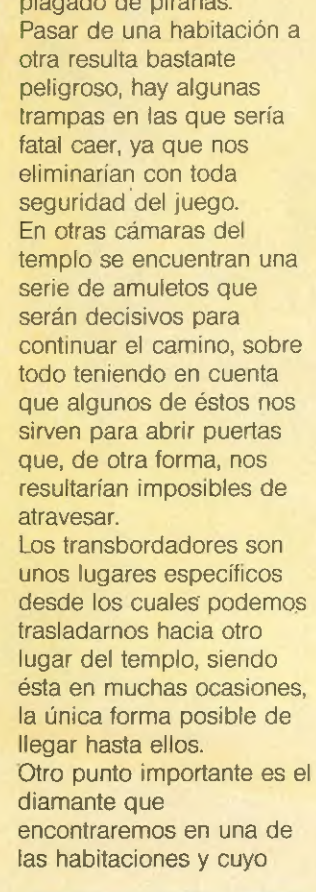
ABU SIMBEL PROFANATION

Dinamic
48 K
Tipo de juego: Videoaventura
P.V.P.: 2.100

Con este programa culmina la trilogía que comenzara con Saimazon y que narra las aventuras y desventuras de Johny Jones, el supuesto hermano de Indiana Jones.
En esta ocasión, el



personaje en cuestión se ha visto poseído por la maldición de Abu Simbel, que le ha convertido en una especie de ser deforme.
Para volver a recuperar su antigua apariencia, deberá de viajar hasta el templo de Abu Simbel, donde se encuentra escondido el secreto que le devolverá su personalidad.
Nada más comenzar el juego nos encontramos en la primera de las habitaciones del templo, en donde una enorme estatua se alza ante nosotros guardando la puerta de entrada al misterioso templo del antiguo Faraón. Tras ésta, una enorme estructura laberíntica, compuesta por 45 habitaciones, pondrá a

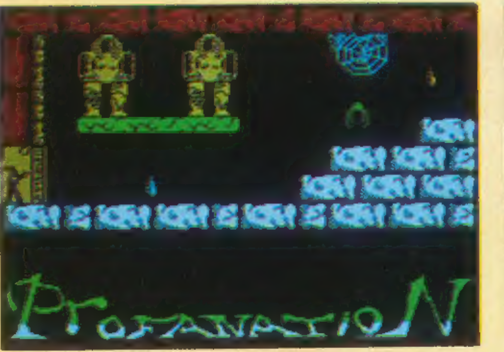


prueba nuestra habilidad y paciencia, que con toda seguridad, en más de una ocasión llegarán a rebasar los límites previstos.
Existen, distribuidas por todas las habitaciones, numerosas trampas que tienen como objetivo evitar a toda costa que lleguemos hasta la tumba del Faraón. Arañas asesinas, serpientes, gotas de ácido, momias, y un sin fin de personajes más nos atacarán continuamente.
En la parte inferior del templo hay incluso una especie de río subterráneo plagado de pirañas.
Pasar de una habitación a otra resulta bastante peligroso, hay algunas trampas en las que sería fatal caer, ya que nos eliminarían con toda seguridad del juego.
En otras cámaras del templo se encuentran una serie de amuletos que serán decisivos para continuar el camino, sobre todo teniendo en cuenta que algunos de éstos nos sirven para abrir puertas que, de otra forma, nos resultarían imposibles de atravesar.
Los transbordadores son unos lugares específicos desde los cuales podemos trasladarnos hacia otro lugar del templo, siendo ésta en muchas ocasiones, la única forma posible de llegar hasta ellos.
Otro punto importante es el diamante que encontraremos en una de las habitaciones y cuyo













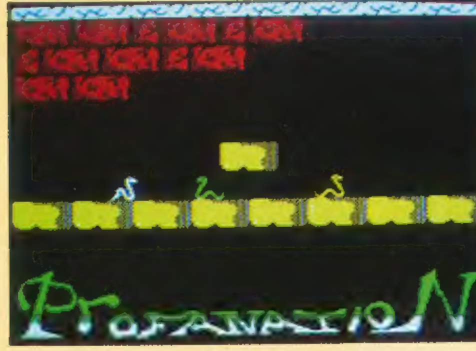


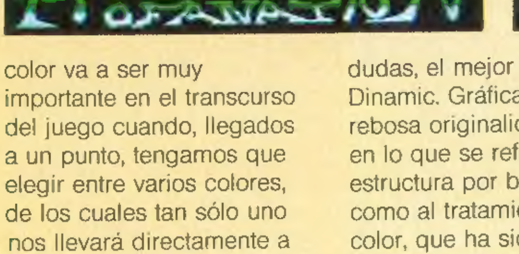












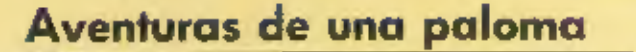
color va a ser muy importante en el transcurso del juego cuando, llegados a un punto, tengamos que elegir entre varios colores, de los cuales tan sólo uno nos llevará directamente a la cámara del Faraón.
Valoración. Es, sin lugar a dudas, el mejor juego de Dinamic. Gráficamente, rebosa originalidad, tanto en lo que se refiere a la estructura por bloques, como al tratamiento del color, que ha sido cuidado a la perfección.
Cada una de las pantallas por las que tenemos que ir pasando, han sido estudiadas al máximo para conseguir de ese modo, crear una estructura de alto nivel de dificultad que sólo puede superarse examinando a fondo cada uno de los movimientos de los objetos y de los personajes, lo que va a suponer al jugador tener que dedicarle bastante tiempo a cada uno de las pantallas.
Hay dos tipos de movimiento, ambos en forma de salto, uno más largo y otro más corto. Cada uno de éstos nos servirá en un determinado momento del juego.
El nivel de dificultad es bastante alto, pero gracias a la vistosidad de los gráficos resulta muy ameno e incluso excitante el hecho de pasar de una pantalla a otra, sobre todo si tenemos en cuenta que éstas son mejores y también más difíciles según vamos avanzando.
Un juego, en definitiva muy bueno, recomendado para maníacos de los videojuegos, que estamos seguros van a ver colmados sus deseos de superación, con el aliciente, además, de ese premio de 50.000 ptas. para el primero que consiga llegar al final.

Originalidad	* * * *
Gráficos	* * * * *
Movimiento	* * * *
Sonido	* * * *
Valoración	* * * * *

1	*Knigt Lore*	Ultimate
2	*Alien 8*	Ultimate
3	*Wally*	Mikro gen
4	*Underwurlde*	Ultimate
5	*Decathlon*	Ocean
6	*Match Day*	Ocean
7	*Babaliba*	Dinamic
8	*Gihft from the goods*	Ocean
9	*Cyclone*	Vortex
10	*Pyjamarama*	Mikro Gen
11	*Jet Set Willy*	Software Projects
12	*Match Point*	Psion
13	*Zaxxon*	U.S. Gold
14	*Saimazoon*	Dinamic
15	*Sabre Wulf*	Ultimate
16	*Tir Na Nog*	Gargoyle
17	*Avalon*	Hewson Consult.
18	*Raid Over Moscow*	U.S. Gold
19	*Rocky*	Dinamic
20	*Bruce Lee*	U.S. Gold

Esta lista ha sido confeccionada con la opinión de nuestros lectores. En un futuro esperamos la colaboración de muchos más, los cuales podrán emitir sus opiniones llamándonos directamente al Tlf. 654 32 11 (con el prefijo 91 para los de fuera de Madrid). También podrá hacerse por carta indicando en el sobre MICRO HITS.

Aventuras de una paloma



POTTY PIGEON







Gremlin Graphics/Serma

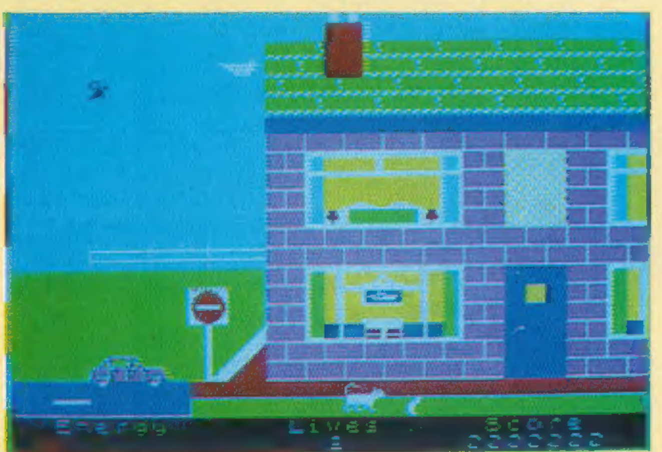
48 K

Tipo de juego: Arcade

P.V.P.: 1.900

Gremlim Graphics es una compañía especializada en la creación de personajes del mundo animal, y en someterlos a las más duras pruebas de subsistencia. En esta ocasión el protagonista es Potty Pigeon, que en castellano viene a ser algo así como «Paloma Loca».

Y es precisamente esa paloma la que tenemos que dirigir en su arriesgada aventura por un mundo hostil, lleno de peligros que pondrán a prueba nuestra capacidad de reacción ante las situaciones difíciles.





En la primera pantalla que aparece al principio del juego, vemos a la paloma en el suelo, y sobre las ramas de un árbol, a unos polluelos hambrientos a los que tenemos que alimentar con los gusanos que se encuentran repartidos por las diferentes pantallas que forman este juego.

Cada vez que cojamos unos gusanos, tendremos necesariamente que ir a otra pantalla para conseguir nuevos alimentos superando una serie de peligros distintos: arañas, pájaros asesinos, flores muy peligrosas, helicópteros, automóviles y multitud de edificios.

En la parte inferior de la pantalla se encuentra un medidor de energía que será fundamentalmente durante el transcurso del juego. Cuando la paloma vuela, ésta disminuye su tamaño notablemente y sólo podremos recuperarla volviendo al suelo o comiendo unos insectos negros que se encuentran en algunas pantallas.

Para que todo no sea complicado, disponemos de un mecanismo de defensa que consiste en lanzar una especie de huevos con los que podemos neutralizar a nuestros enemigos durante algunos segundos, lo que nos permitirá desenvolvernos con una mayor libertad de movimientos.

Según vayamos recogiendo la comida de las pantallas más cercanas, tendremos que ir cada vez más lejos a por nuevos alimentos. Sin embargo, existe una posibilidad de simplificar el riesgo, y es la de tomar la dirección contraria, ya que desde la primera pantalla se puede salir tanto hacia la derecha como hacia la izquierda, lo que lógicamente, acorta el recorrido de las pantallas más lejanas. Esto ocurre porque el juego está planificado como si diera una vuelta en círculo y regresara al mismo lugar.

Si logramos completar el recorrido y recoger todo el alimento, comenzará una nueva fase con un nivel de dificultad más alto y por supuesto con nuevos peligros.

Valoración. El programa está en la línea del resto de las producciones de la compañía, que ha dotado a todos sus juegos de una personalidad gráfica propia que los hace fácilmente reconocibles. Al igual que ocurre con otros personajes creados por Gremlim Graphics, el protagonista del juego es de reducidas dimensiones (ocupa cuatro caracteres), mientras que los gráficos de edificios son bastante grandes, creando una cierta desproporción que además de curiosa, resulta muy interesante en este tipo de programas.

Es un juego no muy complicado, con buenos detalles y un buen nivel de adicción.

Originalidad	* * *
Gráficos	* * *
Movimiento	* * *
Sonido	* * *
Valoración	* * *

El botín del grumete

BOOTY

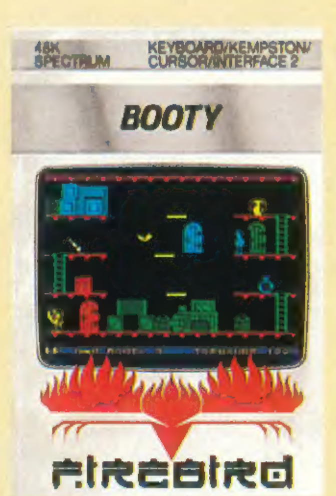
Firebird/ABC

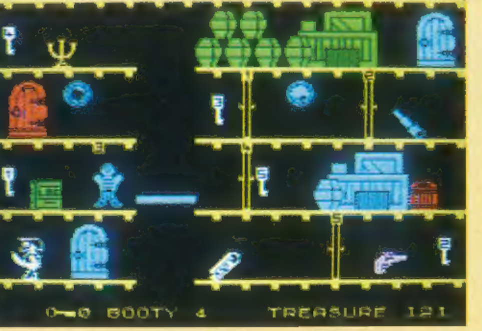
48 K

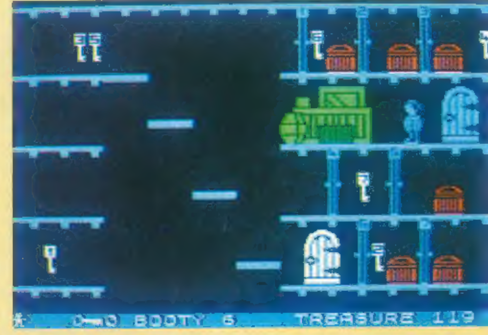
Tipo de juego: Arcade

P.V.P.: 795

Aunque este juego no acaba de salir al mercado, merece nuestra atención tanto por la calidad del programa en sí como por el precio.







Pertenece a la serie antigua de la compañía Firebird, que en la actualidad ha dirigido su mirada hacia el terreno galáctico con programas como Buggy Blast o el conocido Gyron.

En Booty, estamos muy lejos de los escenarios galácticos. La acción se desarrolla en los camarotes de un barco, donde Jim, el protagonista del juego, recorre las salas bajo la cubierta en busca de tesoros escondidos a lo largo del barco, los cuales están celosamente guardados por unos peligrosos piratas que recorren los pasillos de las bodegas con el sable en la mano.

Todas las habitaciones del barco están cerradas y sólo pueden ser abiertas con las llaves correspondientes, que se encuentran numeradas. Algunas de éstas son muy difíciles de conseguir y otras, aunque en un principio son más fáciles, luego plantean problemas ya que si tenemos una llave y pasamos por el lugar donde se encuentra otra, la primera se cambiará por la segunda con el consiguiente perjuicio para nosotros que tendremos que ir a buscarla nuevamente.

Una vez que hemos recogido todos los objetos de las 20 bodegas del Black Galleon's, tenemos 45 segundos para encontrar la llave de bronce que se halla en un montón junto al botín. Si logramos por segunda vez recoger todas las partes del botín, podremos coger la llave de plata y si lo logramos una tercera vez, la llave de oro.

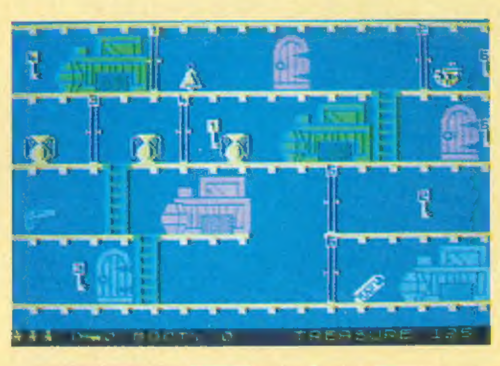
Cada vez que completemos una fase, la velocidad y las complicaciones aumentarán con respecto a la anterior.

Los peligros que nos acechan son muchos: bombas que estallan de repente, el loro del capitán, un peligroso fantasma, puertas que dan al mar y muchos más.

Valoración. Se trata de un programa muy entretenido, con buenos gráficos en el que quizás el movimiento esté algo anticuado con respecto a las técnicas que se utilizan hoy día; pero a pesar de ello, no desmerece en absoluto el resto del juego.

El nivel de dificultad es lo suficientemente alto como para mantener nuestro interés pero sin rayar en lo imposible. Es un juego con los ingredientes necesarios para mantener nuestra atención bastantes horas.

Originalidad	* * *
Gráficos	* * *
Movimiento	* * *
Sonido	* * *
Valoración	* * *





CODIGO "ASCII"	GDU	
144	"A"	65368
145	"B"	65376
146	"C"	65384
147	"D"	65392
162	"S"	65512
163	"T"	65520
164	"U"	65528

Direcciones de comienzo «GDU» (48 K).

Definición

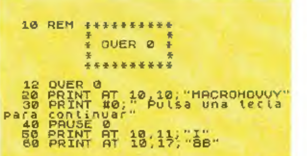
La sentencia «OVER» controla la *sobreimpresión* de los caracteres. Su estructura general es:

SENTENCIA	ARGUMENTO
OVER	código de control

Ejemplos:
- PRINT OVER 1; «BASIC»
- OVER Ø
- OVER c
- PRINT OVER n; 1

Cuando el código de control es «cero», al visualizar un carácter en una posición de la pantalla, se borra el que anteriormente hubiese en esa misma posición. Al conectar el ordenador al código de control por defecto es el «Ø».

Ejemplo:



```
10 REM ***********
       *         *
       * OVER 0  *
       *         *
       ***********
12 OVER 0
20 PRINT AT 10,10;"MACROHOVVY"
30 PRINT #0;" Pulsa una tecla para continuar"
40 PAUSE 0
50 PRINT AT 10,11;"I"
60 PRINT AT 10,17;"8B"
```

Con el código de control «uno», se combinan uno a uno, los pixel del carácter antiguo y del nuevo, de acuerdo con la función lógica «XOR» (exclusive OR), cuya *tabla de verdad* es la siguiente:

a	b	XOR
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	Ø

Es decir, si los dos pixel que se combinan tienen color de «papel» o de «tinta» el pixel resultante tendrá color de «papel»; por el contrario, si son distintos, es decir, si uno tiene color de papel y otro de tinta, el pixel resultante tendrá color de «tinta».

Ejemplo:

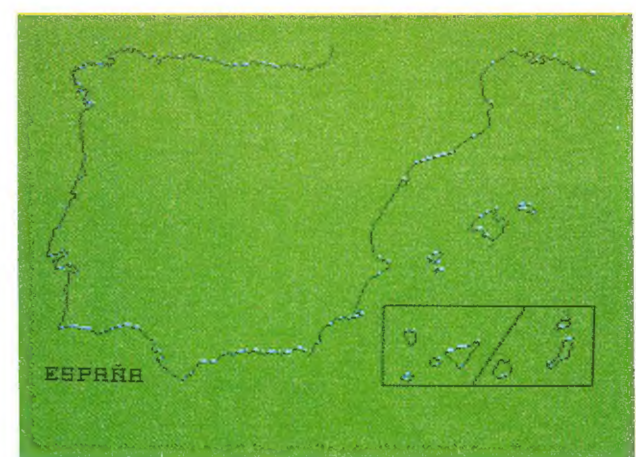
```
10 REM *************
       *           *
       * A OVER X  *
       *           *
       *************
20 PRINT AT 0,0;" A over X = "
30 PRINT AT 0,12;"A"
40 PRINT OVER 1;AT 0,12;"X"
```

Si el código de control es distinto de los mencionados anteriormente nos aparecerá el error de «Invalid colour» o «Interger out of range».

El siguiente programa es una aplicación curiosa de la sentencia «OVER».

```
10 REM    ***********
          *         *
          * "OVER"  *
          *         *
          ***********
20 REM   © MICROHOBBY
30 LIST : OVER 1
40 PRINT #0; INVERSE 1;" Pulsa una tecla para continuar ": PAUSE 0: INPUT 0
50 FOR n=1 TO 3
60 PRINT BRIGHT 1;AT n,13;"
"
70 NEXT n
80 PRINT INVERSE 1;AT 6,10;"
"
90 OVER 0
```





Al realizar gráficos con la sentencia «OVER» activada, se obtienen unos resultados artísticos.

El siguiente programa realiza unos círculos crecientes, tomando como origen las cuatro esquinas; aproximadamente tarda dos minutos en completar el gráfico.

```
10 REM ************
       *          *
       * OVERCIRC *
       *          *
       ************
20 BORDER 4: PAPER 4: INK 0: C
LS
22 OVER 1
30 FOR n=0 TO 86 STEP 2
40 CIRCLE n,n,n
50 CIRCLE 255-n,n,n
60 CIRCLE 255-n,175-n,n
70 CIRCLE n,175-n,n
80 NEXT n
90 OVER 0
```

El programa número «7», con la función «OVER» activada, genera un punto aleatorio, a partir del cual se trazan líneas rectas hacia los bordes de la zona de visualización, posteriormente se genera otro y se repite el proceso, de esta manera se obtiene una combinación «espectacular»; el color del fondo va cambiando de color, con una temporización, mientras que la «tinta» tiene color de contraste.

SCREEN$

Acceso al teclado

MODO E

LEN SYMBOL SHIFT



SCREEN $

Tipo de sentencia

Función auxiliar.

Definición

Esta función retorna una cadena con el carácter existente en una determinada posición. Su estructura general es la siguiente:

SENTENCIA	ARGUMENTO
SCREEN$	(fila, columna)

Ejemplo:
- PRINT SCREEN$ (5,7)
- IF SCREEN$ (2, 3) = "9" THEN...
- LET a$ = SCREEN$ (1O, 11)
- LET b = VAL (SCREEN$ (7, 4))

«SCREEN$» reconoce cualquier carácter comprendido entre los códigos 32 (espacio) y el 127 (©) en decimal; independientemente del atributo temporal que tengan.

Ejemplo:

```
10 PRINT FLASH 1; AT 10, 10;
"HOP"
20 PRINT SCREEN$ (10, 11)
```

Esta función retorna una cadena vacía cuando la posición especificada contiene un carácter fuera del rango.

Ejemplo:

```
10 PRINT CHR$ 135
20 PRINT SCREEN$ (0, 0)
```

El programa número «8» genera un laberinto y utiliza la función «SCREEN$» para detectar los muros y los objetos.

Almacenamiento de pantallas

«SCREEN$» puede utilizarse conjuntamente con las sentencias «SAVE» y «LOAD» para grabar en cinta y posteriormente cargar la imagen (gráficos + texto) que hay en pantalla.

La estructura general es:

a) Salvar pantallas.

SAVE nombre SCREEN$

donde el nombre es una cadena entrecomillada de un máximo de diez caracteres.

Ejemplo:

SAVE «datalla» SCREEN$

b) Cargar pantallas.

LOAD nombre SCREEN$

Ejemplo:

LOAD «datalla» SCREEN$

La sentencia «VERIFY» no opera conjuntamente con «SCREEN$», de manera que no puede verificarse la grabación.

POINT

Acceso al teclado



MODO E

POINT SYMBOL SHIFT

Tipo de sentencia

Función auxiliar

Definición

La función «POINT» indica si un pixel es de color de «tinta» o de «papel».

Su estructura general es:

SENTENCIA	ARGUMENTO
POINT	(coord. X, coord. Y)

Ejemplos:
- PRINT POINT (12, 12O)
- LET a = POINT (7O, 4O)
- IF POINT (5, 3) = 1 THEN...
- PRINT POINT (X, Y)

La función «POINT» retorna un «cero» si el pixel especificado tiene color de «papel» y un «uno» si es de color de «tinta».

Ejemplo:

a)

```
PLOT 10, 120
PRINT POINT (10, 121)
```

b)

```
PLOT 60, 127
PRINT POINT (60, 127)
```

Programa

La estructura del programa «LABERINTO», el número 8, es la siguiente:

10	: Comentario con el nombre del programa.
12	: Asignación de los colores (azul) para borde y fondo, y (blanco) para los caracteres.
20	: Llamada a la subrutina de visualización de instrucciones.
22	: Asignación a la variable (detecta) del número de línea donde comienza la subrutina de detección.
24-28	: Asignación de los valores iniciales de las variables (contador), (tiempo) y (record).
40-60	: Rellenar el área de visualización con los muros del laberinto, utilizando el símbolo de (copyright) (c) en inversa y tinta verde.
80-160	: Visualización de las habitaciones, dentro del muro, utilizando el carácter espacio.
180-210	: Bucle para la lectura y visualización de los caminos.
220-270	: Tabla con los datos correspondientes a las coordenadas (X) e (Y) de los caminos.
290	: Visualización de los sacos, utilizando el símbolo del dólar con atributo temporal de (FLASH).
310-340	: Generación aleatoria de la figura, en cualquiera de las posiciones de las líneas Ø o 21.
350	: Visualización de la figura a desplazar utilizando uno de los símbolos predefinidos.
354-358	: Visualización de los tres marcadores del juego: – Tiempo transcurrido. – Sacos recogidos. – Puntuación máxima.
380-410	: Comprobación de la tecla pulsada, asignación de las nuevas coordenadas y llamada a la subrutina (detecta).
412	: Incremento en una unidad del tiempo transcurrido.
14-416	: Visualización del tiempo; presentando las unidades en la misma posición.
418	: Comprueba si ha transcurrido el tiempo máximo y calcula la puntuación obtenida.
419	: Comprueba si se han recogido los cuatro sacos y calcula la puntuación.
5Ø2	: Borrado de la figura.
510	: Visualización de la puntuación obtenida.
520	: Cambia el color del papel y de la tinta, de la línea Ø.
530	: Temporización.
540-580	: Comprueba si se desea jugar otra partida.
590	: Borrado del mensaje visualizado en el canal Ø.
600	: Comprueba si los puntos obtenidos superan la puntuación máxima.
610	: Visualización del nuevo record.
620	: Borrado del mensaje de la línea Ø.
630	: Inicialización de las variables (tiempo) y (contador).
1000	: Comienzo de la subrutina (detecta).
1Ø1Ø-1Ø4Ø	: Comprueban que la figura no se sale de la pantalla.
1Ø5Ø	: Comprueba si la nueva posición es un muro.
1Ø52	: Comprueba si la nueva posición es un saco, y aumenta y visualiza el valor de la variable contador.
1Ø6Ø	: Borra la antigua posición.
1Ø7Ø	: Calcula la nueva posición.
1Ø8Ø	: Visualiza la posición de la figura.
2ØØØ-21ØØ	: Presentación del juego.
211Ø-226Ø	: Presentación de las instrucciones.

GRAFICOS DEFINIDOS

Posiblemente éste sea uno de los capítulos más esperados por el lector, ya que, tras su lectura, le va a permitir definir, sin dificultad, sus propios gráficos, con el único límite de su imaginación.

Abreviadamente se les conoce por las siglas inglesas «UDG», (User Defined Grafic) o las españolas «GDU» (Gráficos Definidos por Usuario).

Con las técnicas que se van a describir en este capítulo, se pueden definir hasta 21 gráficos, quedando asignados a las letras comprendidas entre la «A» y la «U», ambas inclusive.

```
* GDU GENERATOR *
Letras        ABCDEFGHIJKLMNOPQRSTU
Graficas
Otros efect
←↓↑→ para mover          12345678
ENTER para llenar       1
DELETE para borrar      2
V Borrar todo GDU       3
I Para invertir         4
M Espejo ↑/↓            5
N Espejo ←/→            6
T Volver                7
X Salida                8
Q Recargar caracteres del juego
S Registrar en CINTA estos GDU
L Cargar estos GDUs de la CINTA
Oprimir SPACE para pasat a
proxima GDU
```

¿Cómo se almacenan?

Al igual que los caracteres, cada «GDU» está formado por una matriz de «8» por «8» pixel, en total 64; el contenido de cada uno de éstos (1-activo, Ø-desactivo) queda reflejado en un *bit*, que es la unidad de información más pequeña utilizada en Informática.

Estos «*bits*» agrupados de «8» en «8» es lo que se denomina *byte*; en el caso de los gráficos, un «byte» queda formado por los «8» pixel de una línea, por tanto para almacenar un «GDU» se necesitan ocho «bytes» de información, y para almacenar los 21 posibles gráficos son necesarios 168 «bytes» de información.

Dentro de la memoria del Spectrum, existe una zona destinada al almacenamiento de los 168 «bytes»; ésta se encuentra localizada al final de la misma.

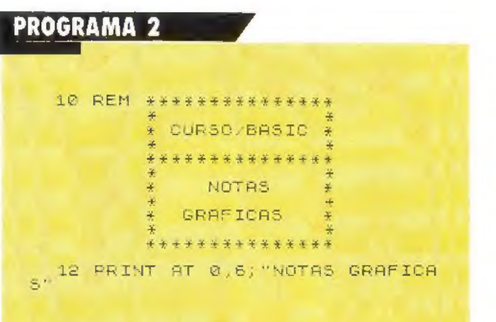
Para almacenar un «byte» es necesario indicar en qué posición de memoria debe efectuarse. La sentencia encargada de escribir datos en la memoria es «POKE», su estructura general es:

SENTENCIA	ARGUMENTO
POKE	dirección, dato

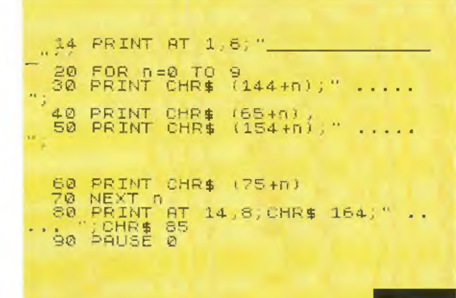
PROGRAMA 1

```
  1 REM ***************
        *             *
        * CURSO/BASIC *
        *             *
        ***************
        *             *
        *  ALFABETO   *
        *             *
        *  ESPAÑOL    *
        *             *
        ***************
  2 PRINT #0;AT 1,2;"Espera un
momento por favor"
  5 RESTORE
 10 FOR g=USR "a" TO USR "j"+7
 20 READ dato
 30 POKE g,dato
 40 NEXT g
 50 DATA 4,8,56,4,60,68,60,0
 60 DATA 4,8,56,68,120,64,60,0
 70 DATA 8,16,0,48,16,16,56,0
 80 DATA 4,8,56,68,68,68,56,0
 90 DATA 8,16,68,68,68,68,56,0
100 DATA 68,0,68,68,68,68,56,0
110 DATA 56,0,120,68,68,68,68,0
120 DATA 24,66,98,82,74,70,66,0
122 DATA 0,16,0,16,32,66,60,0
124 DATA 0,8,0,8,8,8,8,0
126 INPUT 0
130 PRINT #0;AT 0,1;"Ya está, p
ulsa una tecla para",AT 1,10;"ha
cer ""NEW"""
140 PAUSE 0
150 NEW
```

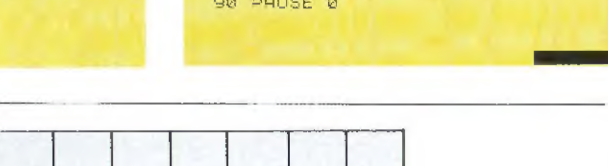

PROGRAMA 2

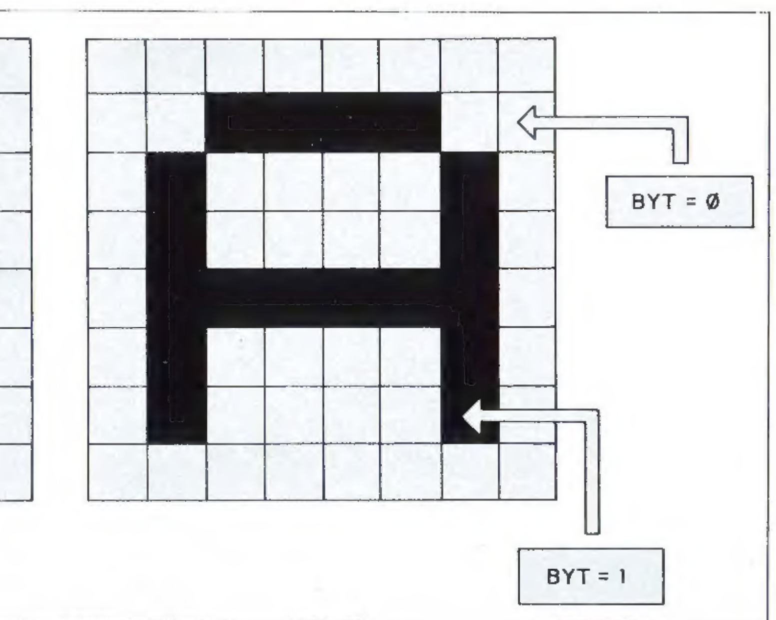


```
10 REM **************
       *            *
       * CURSO/BASIC *
       *            *
       **************
       *            *
       *   NOTAS    *
       *            *
       *  GRAFICAS  *
       *            *
       **************
12 PRINT AT 0,6;"NOTAS GRAFICA
S"
```



```
14 PRINT AT 1,6;"______________"
20 FOR n=0 TO 9
30 PRINT CHR$ (144+n);" ....."
40 PRINT CHR$ (65+n),
50 PRINT CHR$ (154+n);" ....."
60 PRINT CHR$ (75+n)
70 NEXT n
80 PRINT AT 14,8;CHR$ 164;" ..
.";CHR$ 65
90 PAUSE 0
```





Bit y Byte.

El siguiente programa nos indica cuál es la dirección de comienzo de cada «GDU»

```
 1 REM ****************
       *              *
       * DIRECCIONES  *
       *              *
       *   GRAFICOS   *
       *              *
       ****************
 2 LET letra=65
10 FOR n=USR "a" TO USR "u" ST
EP 8
20 PRINT CHR$ letra;" ........
......"
30 PRINT n
40 LET letra=letra+1
50 NEXT n
```

Como es lógico, estas direcciones son distintas en los Spectrum de 16 K y 48 K, ya que este último tiene más memoria, por tanto, si para definir el primer «byte» del «GDU» asignado a la letra «A» utilizamos:

POKE 65368, dato

esta fórmula será correcta para un Spectrum de 48 K, pero no para uno de 16 K; por lo tanto, para almacenar los gráficos debe utilizarse la expresión:

USR (letra)

que lo almacena en su posición correspondiente, independientemente de la capacidad d memoria del ordenador. Utilizando esta expresión, el ejemplo anterior que-

Rutina movimiento. Programa laberinto.

Subrutina detecta. Programa laberinto.

COMPUTIQUE

Te da más

10% Dto.

Y también SPECTRUM 48K por sólo 31.500

por solo 42.000 pts

INVESTRONICA

GARANTIA

COMPRAS A PLAZOS HASTA 12 MESES

al comprar tu spectrum

te regalamos









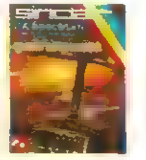












COMPUTIQUE
Abrimos sábados por la tarde

Embajadores, 90
28012 Madrid
Tfno. 2270980

INTERFACE MONITOR

NOVEDAD INFORMAT 85

SALIDA VIDEO COMPUESTO STANDARD

SE CONECTA DIRECTAMENTE AL BUS DE EXPANSIONES

BUS POSTERIOR PARA OTROS PERIFERICOS

3900 ptas

SPECTRUM

LAPIZ OPTICO

4750 ptas

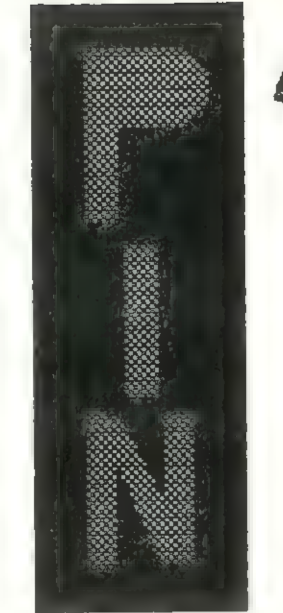
3 NIVELES DE ZOOM
MAS DE 40 FUNCIONES
CASSETTE Y MICRODRIVE
UN SISTEMA DE DIBUJO
A UN PRECIO INCREIBLE



IF SONIDO POR TU TV

3500 ptas

COMPATIBLE JOYSTICK
CONSIGUE MAS REALISMO





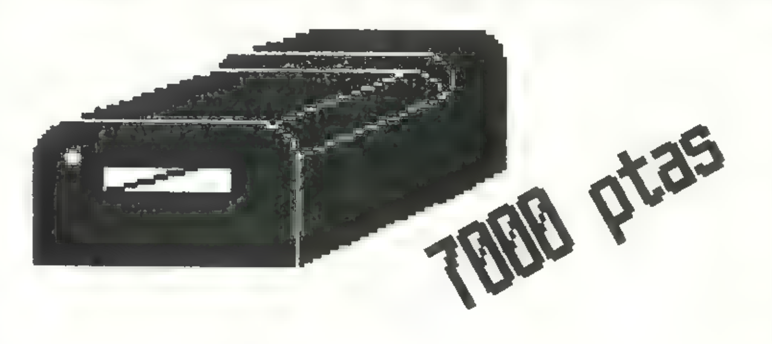
le sacamos jugo
al fruto de la
investigacion

impresora calidad margarita
SEIKOSHA SP-800
con tratamiento de textos
(copy grafico incorporado)
alimentador hoja a hoja
69750 ptas

tambien impresora-plotter
RITEMAN F+
69000 ptas

NECESITAMOS REPRESENTANTES PARA TODA ESPAÑA

MDS tm
MICRODRIVE DISK SYSTEM
SISTEMA OPERATIVO DE DISCO CON MICRODRIVE



7000 ptas

ficheros aleatorios
tiempo medio de acceso 4 seg
amplia la memoria de tu
spectrum hasta 90 k
nuevos comandos basic
al fin tu microdrive funcionara
como un disco de verdad

ven a vernos o pidelo por correo a

PIN SOFT
PASEO DE GRACIA 11 ESC C 2 4
08007 BARCELONA TEL 93 318 24 53

ALGORITMOS DE ORDENACION (y IV)

Javier ALAMAN

En este cuarto y último capítulo sobre Ordenación tratamos sobre el método denominado «QUICKSORT», uno de los más veloces y especialmente recomendado para la ordenación de gran cantidad de datos.

Este es un algoritmo en el que lo más complicado de entender no es su funcionamiento, sino el motivo, casi mágico, de su increíble rendimiento. Su velocidad es tan espectacular que su distribuidor C.A.R. Hoare lo bautizó con el nombre de «quicksort», es decir, ordenación rápida.

Su principio de funcionamiento es el siguiente: tomamos el array y conseguimos que presente una estructura peculiar, en la que el elemento central del array (el n/2) no tiene ningún elemento inferior detrás suyo ni ninguno superior delante. Esto se hace empezando, por ejemplo, por el extremo izquierdo del array, y buscando un elemento que sea superior al que ocupa el centro del array. Si se encuentra, se busca entonces desde el extremo derecho un elemento que sea inferior al central. Cuando éste es obtenido, se intercambian ambos. Este proceso se continúa hasta que efectivamente el elemento central tiene por su izquierda, todo elementos inferiores y por su derecha, todo elementos superiores. (En el proceso, es posible que sea intercambiado el propio elemento central).

En esta situación, lo único que hay que hacer es repetir lo anterior para cada uno de los sub-arrays que quedan a la derecha e izquierda del elemento central. Después, repetir para los sub-arrays de los sub-arrays, y así hasta que éstos consten de un solo elemento, momento en el que, evidentemente, el array estará ordenado.

En la práctica, se plantea el problema de que para cada paso se generan dos sub-arrays que tendrán que ser tratados. Uno de ellos puede serlo inmediatamente, y además, se procura que sea el de mayor tamaño, si lo hay. (Ambos sub-arrays pueden diferir en longitud en una unidad). El otro, tendrá que ser guardado en una «cola de espera» para ser ordenado posteriormente. En lenguajes más avanzados que el BASIC, como pueden ser el PASCAL o el APL, este problema podría resolverse también mediante la recursividad. No nos extenderemos más sobre este punto.

Como ya hemos dicho, éste es el algoritmo que ordena datos con una velocidad mayor, para número de datos a ordenar suficientemente grande. (En el Spectrum, para 100 o más datos, por ejemplo). Sus desventajas, aparte de su mayor complejidad de programación, estriban en que ocupa una cantidad de memoria extra, además de la del propio array (para la cola de trabajos en espera), y en que existen determinadas distribuciones iniciales de datos (muy raras, por otro lado, y muy difícil que se den en la realidad en alguna ocasión) que dan al traste con el sistema, que pasaría a ser para dichos datos el método «lento». De hecho, el caso «malo» se da cuando en cada nuevo sub-array el elemento mayor es precisamente el central.

```
10 REM - quicksort -
15
20 LET m=LN dim/LN 2
30 DIM p(m,2)
40 LET p=1: LET p(1,1)=1: LET p(1,2)=dim
50 LET iz=p(p,1). LET de=p(p,2). LET p=p-1
60 LET i=iz: LET j=de: LET x$=a$(INT ((iz+de)/2),)
70 IF a$(i,)<x$ THEN LET i=i+1 GO TO 70
80 IF x$<a$(j,) THEN LET j=j-1: GO TO 80
90 IF i<=j THEN LET w$=a$(i,): LET a$(i,)=a$(j,): LET a$(j,)=w$: LET i=i+1: LET j=j-1
100 IF i<=j THEN GO TO 70
110 IF j-iz>=de-i THEN GO TO 14█
120 IF i<de THEN LET p=p+1: LET p(p,1)=i: LET p(p,2)=de
130 LET de=j· GO TO 160
140 IF iz<j THEN LET p=p+1. LET p(p,1)=iz: LET p(p,2)=j
150 LET iz=i
160 IF iz<de THEN GO TO 60
170 IF p<>0 THEN GO TO 50
180 RETURN
```

Conclusión

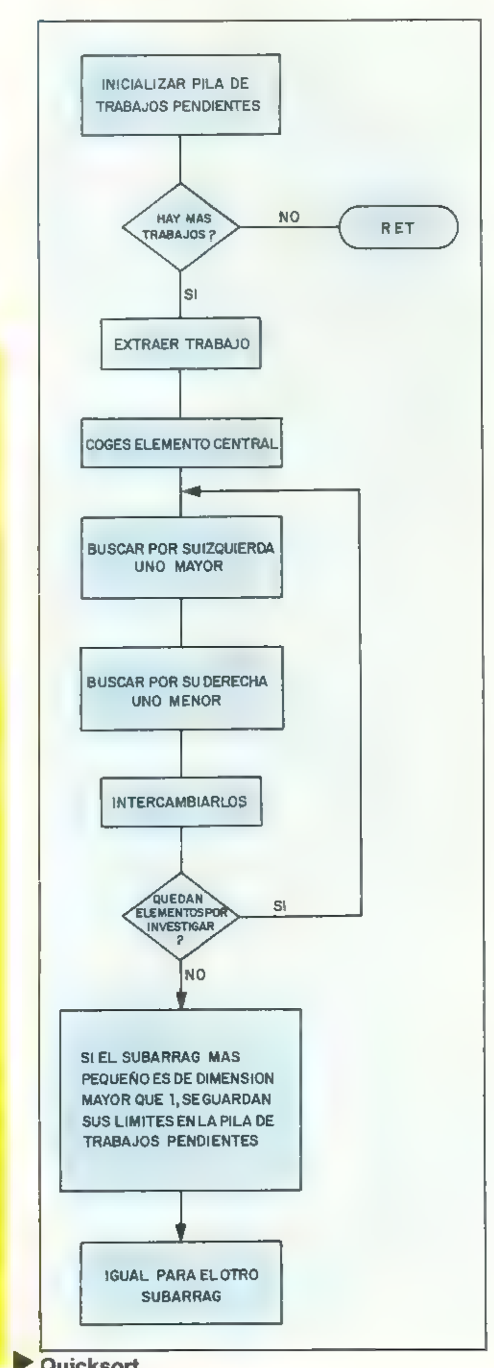
En conclusión, hemos visto que lo que en un principio parecía una tarea simple y trivial, en realidad puede ser objeto de muchos estudios y consiguientes mejoras. En adelante cada vez que tenga usted que ordenar un array de datos, probablemente se lo piense dos veces antes de escoger el método adecuado.

Como hemos visto, para pequeño número de datos (típicamente no más de treinta) los algoritmos elementales son la elección adecuada, ya sea la burbuja o la sacudida para datos semiordenados, ya sea la inserción o la selección para datos sin ningún orden. Para muchos datos, en cambio (más de cien, por ejemplo), el quicksort suele ser la elección más escogida por los programadores profesionales, aunque algunos prefieren el heapsort, que tampoco desmerece, y en cambio es un método más regular en sus resultados, como ya hemos comentado. Aquí, como en muchas otras cosas, es cuestión de gustos.

El lector avispado probablemente se habrá dado cuenta de que queda una franja de valores sin cubrir en las anteriores recomendaciones. En estos valores (entre 30 y 100, típicamente) no hay más remedio que armarse de valor y experimentar con los distintos métodos hasta encontrar el que mejor se ajuste al problema. Para facilitar dicha experimentación, suministramos a continuación un programa que podría titularse «banco de pruebas para algoritmos de ordenación».

Una vez ejecutado RUN aparece un menú con los distintos comandos posibles. Lo primero que hay que hacer es utilizar el comando «A» para definir la aletoriedad de los datos, en unidades de tanto por ciento. Así, un valor de 100 representa unos datos iniciales completamente desordenados, mientras que un valor de 0 corresponde a los datos iniciales completamente ordenados. Luego se utiliza el comando «D» para fijar el tamaño del array. El ordenador generará entonces internamente el array con las especificaciones requeridas. Ahora se podrá probar cuánto





▶ Quicksort

PROGRAMA DE ORDENACION

```
1 DEF FN t()=INT (((PEEK 23672)+256*(PEEK 23673)+65536*(PEEK 23674))*.02)
2 LET g$="": DIM t(10): DIM l$(32)
4 DATA 22,2,11,17,2,16,7,127,32,88,46,65,108,97,109,97,110,22,5,3,17,7,16,0
5 LET x$="": FOR i=1 TO 24: READ vi: LET x$=x$+CHR$ vi: NEXT i: LET vi=1: LET com=0
6 GO SUB 30█
8 IF NOT com THEN LET a$=INKEY$: PRINT AT 15,3;"c"
10 IF com THEN LET a$=c$(1): LET c$=c$(2 TO ): IF c$=CHR$ 13 THEN LET com=0
20 IF SCREEN$ (2,20)=a$ THEN BEEP 1,20: LET rut=3500: GO SUB 200: LET t(8)=tiempo. GO SUB 400. BEEP 2,10
25 IF a$="i" THEN BEEP 1,20. LET rut=3000· GO SUB 200 LET t(7)=tiempo GO SUB 400: BEEP 2,10
27 IF a$="g" THEN BEEP 1,20. LET g$=g$+CHR$ 88 FOR i=1 TO 10 LET g$=g$+STR$ t(i)+"$": NEXT i· BEEP 2,10
30 IF a$="b" THEN BEEP 1,20 LET rut=4000. GO SUB 200 LET t(2)=tiempo GO SUB 400 BEEP 2,10
32 IF a$="8" THEN BEEP .1,50: LET vi=vi+1· GO SUB 500: LET vi=vi-1· IF i<>0 THEN LET vi=vi+1. FOR j=1 TO 10. GO SUB 550: LET t(j)=numero: NEXT j: GO SUB 400 BEEP .2,10
33 IF a$="5" THEN BEEP .1,50· IF vi>1 THEN LET vi=vi-1: GO SUB 500 IF i<>0 THEN FOR j=1 TO 10: GO SUB 550: LET t(j)=numero: NEXT j. GO SUB 400: BEEP .2,1█
35 IF SCREEN$ (2,16)=█$ THEN BEEP 1,20 LET rut=4500. GO SUB 200: LET t(4)=tiempo: GO SUB 400 BEEP 2,10
40 IF a$="s" THEN BEEP 1,20: LET rut=5000. GO SUB 200: LET t(6)=tiempo· GO SUB 400: BEEP 2,10
45 IF a$="c" THEN BEEP 1,20: LET rut=5500: GO SUB 200: LET t(5)=tiempo: GO SUB 400: BEEP 2,10
50 IF a$="h" THEN BEEP 1,20: LET rut=8000: GO SUB 200: LET t(3)=tiempo: GO SUB 400: BEEP 2,10
55 IF a$="q" THEN BEEP 1,20: LET rut=6000: GO SUB 200: LET t(1)=tiempo GO SUB 400· BEEP 2,10
60 IF a$="d" THEN BEEP 1,20: GO SUB 130: BEEP 2,10
65 IF SCREEN$ (2,19)=a$ THEN BEEP 1,20: PRINT AT 19,6; PAPER 1; INK 7;"dame la aleatoriedad"· GO SUB 900 LET t(9)=numero: BEEP 2,10: GO SUB 400
70 IF SCREEN$ (2,18)=a$ THEN BEEP 1,20 INPUT "Dame los comandos: ";c$: LET c$=c$+CHR$ 13: GO SUB 300 PRINT AT 15,3; FLASH 1;"c": LET com=1
100 GO TO 8
110 FOR i=1 TO dim PRINT a(i);" ", NEXT i· PRINT · RETURN
120 FOR i=1 TO dim LET a(i)=b(i): NEXT i RETURN
125 REM - generador de vector -
130 FOR i=1 TO 8· LET t(i)=0· NEXT i: PRINT AT 19,7; PAPER 1, INK 7;"dame la dimension" GO SUB 900· LET t(10)=numero: LET dim=numero: DIM b(dim)· DIM a(dim)· IF t(9)>=100 THEN FOR i=1 TO dim · LET a(i)=INT (RND*90)+10: LET b(i)=a(i). NEXT i: GO SUB 400: RETURN
140 FOR i=1 TO dim: LET a(i)=i: NEXT i
150 FOR i=1 TO dim*.005*t(9)
160 LET j=INT (RND*dim)+1: LET k=INT (RND*dim)+1
170 LET x=a(j): LET a(j)=a(k): LET a(k)=x
180 NEXT i FOR i=1 TO dim: LET b(i)=a(i): NEXT i: GO SUB 400
190 RETURN
195 REM - rut auxiliar -
200 PRINT AT 19,13; FLASH 1;"ESPERA". GO SUB 120: LET t1=FN t(): GO SUB rut: LET t2=FN t(): LET tiempo=t2-t1: IF tiempo<0 THEN LET tiempo=tiempo-1+2↑24
210 RETURN
300 REM - pantalla -
310 BORDER 5 PAPER 5: CLS
320 PAPER 2 PRINT AT 2,1,l$( TO 30),AT 17,1;l$( TO 30) FOR i=2 TO 17: FOR j=1 TO 30 STEP 29: PRINT AT i,j," ": NEXT j· NEXT i
330 PAPER 7: FOR i=3 TO 16: PRINT AT i,2,l$( TO 28)· NEXT i
350 FOR i=5 TO 14. FOR j=10 TO 21: PRINT AT i,j,".": NEXT j: NEXT i
360 PRINT AT 3,23;"sgdos.",x$;"Quicksort";AT 6,3;"Burbuja";AT 7,3;"Heapsort",AT 8,3;"shellsort";AT 9,3,"saCudida",AT 10,3,"Seleccion directa";AT 11,3;"Insercion directa";AT 12,3;"iNsercion binaria",AT 13,3;"Aleatoriedad(%)";AT 14,3,"Dimension";AT 15,3;"coMandos,Guardar,cursores"
370 GO SUB 400
390 RETURN
400 REM - pinta valores -
410 FOR i=5 TO 14· PRINT AT i,23;l$( TO 6),AT i,23;t(i-4): NEXT i. PRINT AT 19,2; PAPER 6,l$( TO 2█): RETURN
50█ REM - busca vi-esima # -
51█ LET cont=0: FOR i=1 TO LEN g$
520 IF g$(i)=x$(10) THEN LET cont=cont+1: IF cont=vi THEN RETURN
530 NEXT i. LET i=0: RETURN
550 REM - valor -
560 LET n$="": LET i=i+1
570 IF g$(i)>="0" AND g$(i)<="9" THEN LET n$=n$+g$(i): LET i=i+1: GO TO 570
580 LET numero=VAL n$: RETURN
```

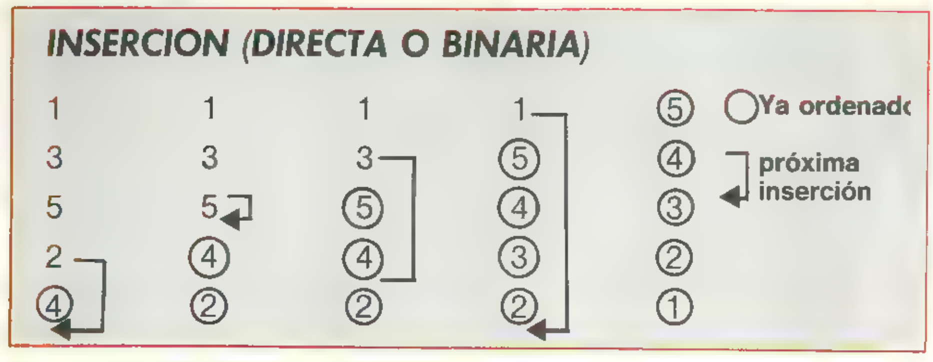
```
 900 REM - saca numero -
 910 IF NOT com THEN INPUT ;nume
ro: RETURN
 920 LET n$=""
 930 IF c$(1)>="0" AND c$(1)<="9
" THEN LET n$=n$+c$(1)  LET c$=c
$(2 TO )· GO TO 930
 940 IF c$=CHR$ 13 THEN LET com=
0
 950 LET numero=VAL n$
 960 RETURN
3000 REM - inserccion directa -
3010 FOR i=2 TO dim
3020 LET x=a(i)
3030 FOR j=i-1 TO 1 STEP -1
3040 IF x>=a(j) THEN GO TO 3070
3050 LET a(j+1)=a(j)
3060 NEXT j
3070 LET a(j+1)=x
3080 NEXT i
3090 RETURN
3500 REM - insercion binaria -
3510 FOR i=2 TO dim
3520 LET x=a(i): LET iz=1: LET d
e=i-1
3530 LET m=INT ((iz+de)/2)
3540 IF x<a(m) THEN LET de=m-1:
GO TO 3560
3550 LET iz=m+1
3560 IF iz<=de THEN GO TO 3530
3570 FOR j=i-1 TO iz STEP -1: LE
T a(j+1)=a(j): NEXT j
3580 LET a(iz)=x
3590 NEXT i
3600 RETURN
4000 REM - burbuja -
4010 LET flag=0: LET k=2
4020 FOR j=dim TO k STEP -1
4030 IF a(j-1)>a(j) THEN LET x=a
(j-1): LET a(j-1)=a(j): LET a(j)
=x: LET k=j: LET flag=1
4040 NEXT j
4050 IF flag THEN LET flag=0: GO
 TO 4020
4060 RETURN
4500 REM - shellsort -
4510 LET t=INT (LN dim/LN 2)-1
4520 FOR m=1 TO t
4530 LET k=-1+2↑(t-m)
4540 FOR i=k+1 TO dim
4550 LET x=a(i)
4560 FOR j=i-k TO k STEP -k
4570 IF x>a(j) THEN GO TO 4600
4580 LET a(j+k)=a(j)
4590 NEXT j
4600 LET a(j+k)=x
4610 NEXT i
4620 NEXT m
4630 RETURN
5000 REM - seleccion directa -
5010 FOR i=1 TO dim-1
5020 FOR j=i+1 TO dim
5030 IF a(i)>a(j) THEN LET x=a(i
): LET a(i)=a(j): LET a(j)=x
5040 NEXT j: NEXT i
5050 RETURN
5500 REM - sacudida -
5510 LET iz=2: LET de=dim: LET k
=dim
5520 FOR j=de TO iz STEP -1
5530 IF a(j-1)>a(j) THEN LET x=a
(j-1): LET a(j-1)=a(j): LET a(j)
=x: LET k=j
5540 NEXT j
5550 LET iz=k+1
5560 FOR j=iz TO de
5570 IF a(j-1)>a(j) THEN LET x=a
(j-1): LET a(j-1)=a(j): LET a(j)
=x: LET k=j
5580 NEXT j
5590 LET de=k-1
5600 IF iz<=de THEN GO TO 5520
5610 RETURN
6000 REM - quicksort -
6010 LET m=LN dim/LN 2
6020 DIM p(m,2)
6030 LET p=1: LET p(1,1)=1: LET
p(1,2)=dim
6040 LET iz=p(p,1): LET de=p(p,2
)  LET p=p-1
6050 LET i=iz: LET j=de: LET x=a
(INT ((iz+de)/2))
6060 IF a(i)<x THEN LET i=i+1: G
O TO 6060
6070 IF x<a(j) THEN LET j=j-1: G
O TO 6070
6080 IF i<=j THEN LET w=a(i): LE
T a(i)=a(j): LET a(j)=w: LET i=i
+1  LET j=j-1
6085 IF i<=j THEN GO TO 6060
6090 IF j-iz>=de-i THEN GO TO 61
20
6100 IF i<de THEN LET p=p+1: LET
 p(p,1)=i: LET p(p,2)=de
6110 LET de=j. GO TO 6140
6120 IF iz<j THEN LET p=p+1: LET
 p(p,1)=iz: LET p(p,2)=j
6130 LET iz=i
6140 IF iz<de THEN GO TO 6050
6150 IF p<>0 THEN GO TO 6040
6160 RETURN
8000 REM - heapsort -
8010 LET wiz=INT (dim/2)+1: LET
wde=dim
8015 IF wiz<=1 THEN GO TO 8035
8020 LET wiz=wiz-1: GO SUB 9000
8030 IF wiz>1 THEN GO TO 8020
8035 IF wde<=1 THEN GO TO 8070
8040 LET x=a(1): LET a(1)=a(wde)
: LET a(wde)=x
8050 LET wde=wde-1: GO SUB 9000
8060 IF wde>1 THEN GO TO 8040
8070 RETURN
8999 REM - criba -
9010 LET i=wiz: LET j=i+i: LET x
=a(i)
9015 IF j>wde THEN GO TO 9060
9020 IF j<wde THEN IF a(j)<a(j+1
) THEN LET j=j+1
9030 IF x>=a(j) THEN GO TO 9060
9040 LET a(i)=a(j): LET i=j: LET
 j=i+i
9050 IF j<=wde THEN GO TO 9020
9060 LET a(i)=x
9070 RETURN
```

SACUDIDA

1	5	5	5
3	1	3	4
5	3	4	3
2	4	2	2
4	2	1	1

BURBUJA

1	5	5	5	5
3	1	4	4	4
5	3	1	3	3
2	4	3	1	2
4	2	2	2	1



SELECCION DIRECTA

1	4	4	4	5
3	3	3	5	4
5	5	5	3	3
2	2	2	2	2
4	1	1	1	1

tarda cada método de ordenación en realizar la tarea. En el menú aparece cada método de ordenación con su nombre en letras minúsculas, salvo una mayúscula. Si apretamos la tecla correspondiente el ordenador se pone a comprobar el tiempo que necesita, expresándolo después sobre la pantalla en segundos. Se pueden probar todos o solo algunos métodos, y en variedad de situaciones de longitud y aletoriedad inicial. Además, se incluyen cuatro comandos de ayuda. Uno de ellos es el comando «M» que permite introducir una serie de comandos seguidos que luego el ordenador ejecutará uno tras otro. Así, para definir una aletoriedad de 100, junto con una dimensión de 50, y para probar el método de la burbuja y el heapsort, el comando m nos permitiría teclear a100d50bh, y después dejaríamos que el Spectrum trabajase él solo en ello. Junto con el comando «M» es importante el comando «G» que guarda para su posterior consulta los datos que en ese momento se ven en la pantalla. Dichos datos podrán ser luego consultados con los cómandos «5» y «8» (sin caps shift), que permitirán ver los datos anteriores o posteriores, respectivamente. Un uso conjunto de dichos comandos sería, por ejemplo, el utilizar el comando «m» para introducir la secuencia a100d10bqgd50bqgd 300bqga0d300bqg que permitiría hacer un estudio comparativo de los métodos burbuja y quicksort para diversas situaciones. (Hay que tener en cuenta que el método de la burbuja para 300 elementos puede llegar a tardar horas.)

TIEMPOS DE EJECUCION DE LOS DISTINTOS METODOS

	DATOS ORDENADOS		DATOS DESORDENADOS	
Inserción directa	12	23	366	1444
Inserción binaria	56	125	373	1327
Selección directa	489	1907	509	1956
Burbuja	5	8	1104	4270
Sacudida	5	9	961	3642
Shellsort	58	116	127	349
Heafsort	116	253	110	241
Quicksort	31	69	60	146

Primera columna: para 256 elementos.
Segunda columna: para 512 elementos.

ASTRO 1

Manolo ORCERA

NOTAS GRAFICAS

A B C D E F G

Spectrum 48 K

Premiado con 15.000 ptas.

En una galaxia desconocida, nos encontramos atrapados en una habitación laberíntica de la que tendremos que salir, sea como sea. Es cuestión de supervivencia.

Por culpa del azar, hemos ido a parar a una galaxia no identificada y hemos caido en manos enemigas. Tras ser encerrados en un inmenso recinto, tendremos que poner todo nuestro empeño en salir de él; pero cuidado, si lo conseguimos no estaremos todavía a salvo. Unos extraños objetos o asteroides, nos atacarán si no repelemos a tiempo su agresión.

Este será, pues, nuestro cometido, y poder acceder a nuestra nave para salir de este «infierno».

Tenemos que conseguirlo...

```
   1 BORDER 1: PAPER 1: INK 7: CLS
   2 GO TO 9900
   3 GO SUB 3000
  10 FOR n=0 TO 30: OUT 254,n: BEEP .01,n: NEXT n: BORDER 1: PAPER 1: INK 7: CLS
  11 GO TO 1000
  20 PRINT AT y1,x1;" ";AT y,x;"$";AT y1+1,x1;" ";AT y+1,x;"¢"
  21 LET x1=x: LET y1=y
  22 LET x=x+2*(INKEY$="p")-2*(INKEY$="o")
  24 LET y=y+2*(INKEY$="a")-2*(INKEY$="q")
  25 IF ATTR (y,x)=13 THEN LET punt=punt+100: RETURN
  26 IF ATTR (y,x)<>15 THEN RETURN
  28 GO TO 20
  29 RETURN
  30 FOR n=0 TO 21 STEP 2: PRINT AT n,0; INK 5; PAPER 2;"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX": NEXT n: READ y2,x2
  31 READ a
  32 IF a<>0 THEN LET y2=y2+(a=2)-(a=1): LET x2=x2+(a=3)-(a=4): GO TO 34
  33 RETURN
  34 PRINT AT y2,x2;" ";AT y2+1,x2;" ": GO TO 31
  40 FOR u=30 TO 25 STEP -1
  42 IF POINT (8*u-4,171-8*y) THEN BEEP .05,0: LET punt=punt+10
  43 NEXT u
  46 PRINT AT y,u; PAPER 0;"▬▬▬▬": BEEP .01,20
  47 PRINT AT y,u; PAPER 0;"    "
  48 RETURN
1000 LET x=30
1010 LET y=2
1020 LET y1=2
1030 LET x1=30
1040 LET punt=0
1050 LET disp=40
2000 REM PATIO
2010 DATA 2,30,4,4,4,4,4,4,4,4,2,2,2,2,1,1,1,1,4,4,4,4,4,4,2,2,2,2,3,3,3,3,3,3,4,4,4,4,2,2,2,2,3,3,3,3,3,3,3,3,1,1,1,1,3,3,3,3,2,2,2,2,2,2,4,4,2,2,4,4,4,4,4,4,4,4,4,4,4
2011 DATA 4,4,4,4,4,4,4,4,4,4,4,1,1,1,1,3,3,3,3,3,3,1,1,1,1,1,1,1,1,4,4,4,4,2,2,4,4,2,2,4,4,2,2,4,4,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,0
2019 RESTORE 2000: GO SUB 30
2020 PRINT AT 18,14; INK 5;"X"
2030 GO SUB 20
2040 FOR n=0 TO 20: OUT 254,n: BEEP .01,n: NEXT n
2100 LET x=31: FOR n=22528 TO 23295-64: POKE n,7*RND: NEXT n: FOR n=0 TO 31: LET o=USR 60000: NEXT n: BORDER 0
2110 FOR k=0 TO 500: PRINT AT 20*RND,0; PAPER 0;"X"
2120 LET o=USR 60000
2130 PRINT AT y,x; PAPER 0;"$";AT y+1,x; PAPER 0;"¢"
2140 LET y=y+(INKEY$="a" AND y<20)-(INKEY$="q" AND y>2)
2150 IF INKEY$="p" THEN PRINT AT y,x-10; PAPER 0;"▬▬▬▬▬▬▬▬▬▬": BEEP .01,45: PRINT AT y,x-10; PAPER 0;"          "
2160 LET f=164-8*y: IF POINT (244,f) OR POINT (244,f+8) THEN FOR n=0 TO 20: OUT 254,n: BEEP .01,n: NEXT n: GO TO 2200
2190 NEXT k
2200 LET punt=punt+k: LET x=30: LET y=2: DATA 2,30,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,1,1,1,1,1,1,1,1,1,1,1
2201 DATA 4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
2202 DATA 1,1,1,1,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0
2210 RESTORE 2200: GO SUB 30
2220 PRINT AT 10,10; INK 5;"X"
2230 GO SUB 20
2240 FOR n=0 TO 20: OUT 254,n: BEEP .01,n: NEXT n
2260 LET x=31: FOR n=22528 TO 23295-64: POKE n,7*RND: NEXT n: FOR n=0 TO 31: LET o=USR 60000: NEXT n: BORDER 0
2270 LET x=31: FOR i=500 TO 0 STEP -1
2280 IF RND>.5 THEN PRINT AT 20*RND,0; PAPER 0;"X"
2290 LET o=USR 60000
2300 LET y=y+(INKEY$="a" AND y<20)-(INKEY$="q" AND y>0)
2310 PRINT AT y,x; PAPER 0;"$";AT y+1,x; PAPER 0;"¢"
2320 IF INKEY$="p" THEN GO SUB disp
2400 NEXT i
2500 GO TO 4000
3000 RESTORE 3000: PLOT 50,170: DATA 30,0,0,-40,-30,0,0,20,30,0,0,-20,20,0,0,10,-15,0,0,10,30,0,-8,0,0,-20,0,20,8,0,0,-20,0,20,15,0,0,-10,-15,0,7,0,8,-10,7,0,0,20,15,0,0,-20,-15,0,50,0,-10,0,0,40,-10,-10,255,255
3010 READ a,b: IF a=255 THEN GO TO 3020
3012 DRAW a,b: GO TO 3010
3020 PRINT AT 9,0
3030 PRINT "MANDOS:"'"Q..ARRIBA"'"A..ABAJO"'"O..IZQUIERDA"'"P..DERECHA Y DISPARO"
3040 PRINT : PRINT "FASE 1.: SOLO SE PERMITE CIRCU  LAR POR EL CAMINO"
3050 PRINT "FASE 2 NO SE PERMITE COLISION   SE TIENE MANDO DE DISPARO"
3060 PRINT "FASE 3 SE PERMITE COLISION PUN- TUA LA DESTRUCCION DE ESTRELLAS"
3070 PRINT #0;"PULSA UNA TECLA": PAUSE 0: CLS : BEEP .5,0: RETURN
4000 FOR n=0 TO 30: OUT 254,n: BEEP .01,n: NEXT n
4010 CLS : PRINT AT 3,2;"TU PUNTUACION HA SIDO DE:"''';TAB 12; FLASH 1;10*PUNT
4020 PRINT AT 15,2;"PULSA S PARA NUEVA PARTIDA"
4030 IF INKEY$<>"s" THEN GO TO 4030
4040 GO TO 10
9899 STOP
9900 DATA 60,94,92,59,12,239,90,99,27,60,112,96,48,32,24,48,129,102,126,60,60,126,102,129,213,192,48,188,23,155,13,142,85,2,13,60,233,216,177,112,14,141,27,151,60,176,64,170,113,176,217,232,61,12,3,171,0
9905 RESTORE 9900: FOR n=USR "a" TO USR "h": READ a: POKE n,a: NEXT n
9910 CLEAR 5e4
9920 DATA 33,255,87,22,0,62,192,6,31,43,94,35,115,43,16,249,114,43,61,32,242,201
9925 RESTORE 9920: FOR n=60000 TO 60021: READ a: POKE n,a: NEXT n
9930 GO TO 3
```



RANSOM

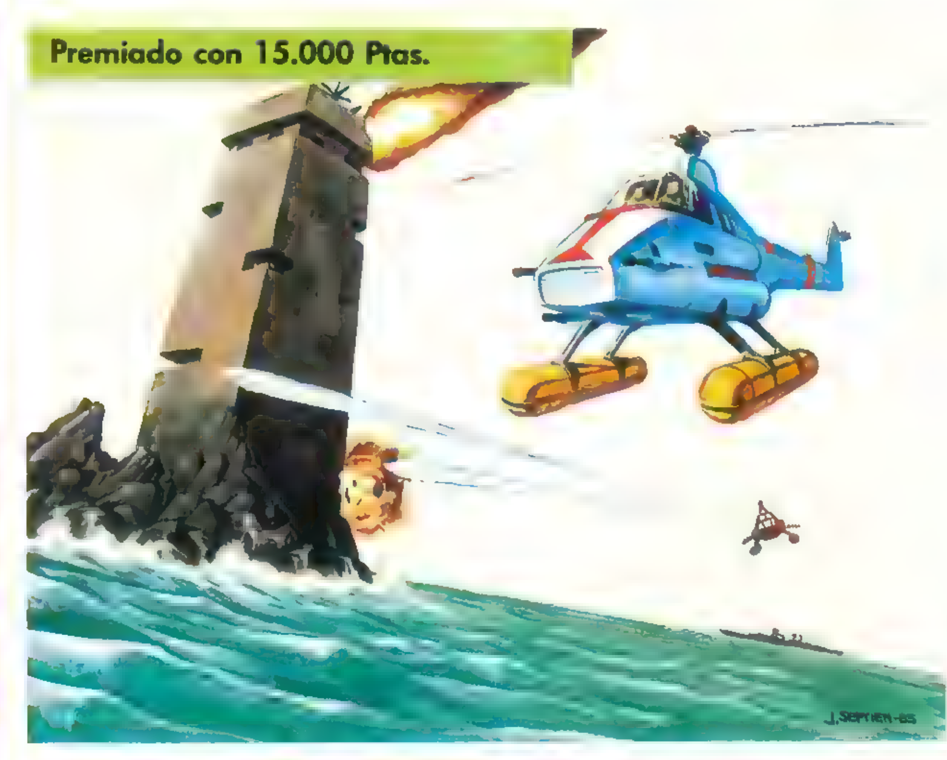
Marco Antonio ARROYO RAMIREZ

NOTAS GRAFICAS

A B C D E F G H I J

Spectrum 48 K

Premiado con 15.000 Ptas.



La misión está definida: rescatar a los integrantes del comando X que han sido capturados por el enemigo y encerrados en la fortaleza conocida por Ransom.

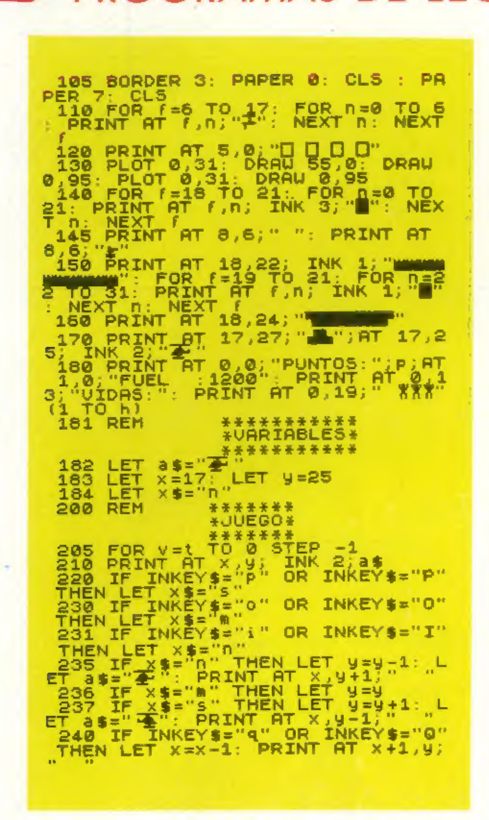
Para ello contamos con un helicóptero con el que sería fácil llegar a la torre donde se encuentran encerrados nuestros compañeros, si no fuera por la vigilancia a que se ven sometidos y la presencia de otro helicóptero que acecha constantemente.

De cualquier modo, hemos de intentar transportarlos hasta el barco que espera en la proximidad esquivando el certero disparo de los cañones que defienden infatigablemente la muralla.

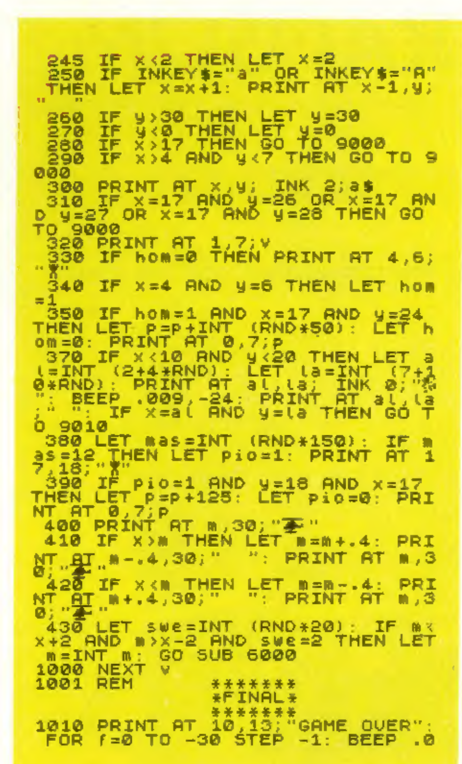
Como podréis comprobar, no será una tarea fácil, sobre todo si nos quedamos sin combustible; pero si lo conseguimos, iremos acumalando puntos por cada hombre rescatado.

Las teclas a utilizar son: P, derecha; O, frenar; I, izquierda; Q, arriba; A, abajo.

```
   1 FOR f=0 TO 7: READ w: POKE USR "a"+f,w: NEXT f
   2 DATA 248,0,0,136,248,0,0,128
   3 FOR f=0 TO 7: READ w: POKE USR "b"+f,w: NEXT f
   4 DATA 255,64,240,248,252,254,32,252
   5 FOR f=0 TO 7: READ w: POKE USR "c"+f,w: NEXT f
   6 DATA 31,0,0,17,31,0,0,1
   7 FOR f=0 TO 7: READ w: POKE USR "d"+f,w: NEXT f
   8 DATA 255,2,15,31,63,127,4,63
   9 FOR f=0 TO 7: READ w: POKE USR "e"+f,w: NEXT f
  10 DATA 255,8,8,8,255,64,64,64
  11 FOR f=0 TO 7: READ w: POKE USR "f"+f,w: NEXT f
  12 DATA 0,0,146,191,255,255,255,255
  13 FOR f=0 TO 7: READ w: POKE USR "g"+f,w: NEXT f
  14 DATA 45,45,30,12,12,18,18,18
  15 FOR f=0 TO 7: READ w: POKE USR "h"+f,w: NEXT f
  16 DATA 255,129,129,129,129,129,129,255
  17 FOR f=0 TO 7: READ w: POKE USR "i"+f,w: NEXT f
  18 DATA 113,176,84,65,139,73,4,96
  19 FOR f=0 TO 7: READ w: POKE USR "j"+f,w: NEXT f
  20 DATA 0,0,96,127,120,32,252,252
  30 LET h=4
  35 LET m=10
  40 LET t=1200
  50 LET hom=0
  60 LET p=0
  65 LET pio=0
  70 GO SUB 5000
 100 REM          ***********
                  *PANTALLA*
                  ***********
```



```
 105 BORDER 3: PAPER 0: CLS : PA
PER 7: CLS
 110 FOR f=6 TO 17: FOR n=0 TO 6
: PRINT AT f,n;"▀": NEXT n: NEXT
 f
 120 PRINT AT 5,0;"□ □ □ □"
 130 PLOT 0,31: DRAW 55,0: DRAW
0,95: PLOT 0,31: DRAW 0,95
 140 FOR f=18 TO 21: FOR n=0 TO
21: PRINT AT f,n; INK 3;"█": NEX
T n: NEXT f
 145 PRINT AT 8,6;" ": PRINT AT
8,6;"▀"
 150 PRINT AT 18,22; INK 1;"▄▄▄▄
▄▄▄▄": FOR f=19 TO 21: FOR n=2
2 TO 31: PRINT AT f,n; INK 1;"█"
: NEXT n: NEXT f
 160 PRINT AT 18,24;"▀▀▀▀▀▀"
 170 PRINT AT 17,27;"▄";AT 17,2
5; INK 2;"▀"
 180 PRINT AT 0,0;"PUNTOS:";p;AT
 1,0;"FUEL   :1200": PRINT AT 0,1
3;"VIDAS:": PRINT AT 0,19;"XXX"
(1 TO h)
 181 REM         ************
                 *VARIABLES*
                 ************
 182 LET a$="▀"
 183 LET x=17: LET y=25
 184 LET x$="n"
 200 REM         *******
                 *JUEGO*
                 *******
 205 FOR v=t TO 0 STEP -1
 210 PRINT AT x,y; INK 2;a$
 220 IF INKEY$="p" OR INKEY$="P"
 THEN LET x$="s"
 230 IF INKEY$="o" OR INKEY$="O"
 THEN LET x$="m"
 231 IF INKEY$="i" OR INKEY$="I"
 THEN LET x$="n"
 235 IF x$="n" THEN LET y=y-1: L
ET a$="▀": PRINT AT x,y+1;" "
 236 IF x$="m" THEN LET y=y
 237 IF x$="s" THEN LET y=y+1: L
ET a$="▀": PRINT AT x,y-1;" "
 240 IF INKEY$="q" OR INKEY$="Q"
 THEN LET x=x-1: PRINT AT x+1,y;
" "
```



```
 245 IF x<2 THEN LET x=2
 250 IF INKEY$="a" OR INKEY$="A"
 THEN LET x=x+1: PRINT AT x-1,y;
" "
 260 IF y>30 THEN LET y=30
 270 IF y<0 THEN LET y=0
 280 IF x>17 THEN GO TO 9000
 290 IF x>4 AND y<7 THEN GO TO 9
000
 300 PRINT AT x,y; INK 2;a$
 310 IF x=17 AND y=26 OR x=17 AN
D y=27 OR x=17 AND y=28 THEN GO
TO 9000
 320 PRINT AT 1,7;v
 330 IF hom=0 THEN PRINT AT 4,6;
"X"
 340 IF x=4 AND y=6 THEN LET hom
=1
 350 IF hom=1 AND x=17 AND y=24
THEN LET p=p+INT (RND*50): LET h
om=0: PRINT AT 0,7;p
 370 IF x<10 AND y<20 THEN LET a
l=INT (2+4*RND): LET la=INT (7+1
0*RND): PRINT AT al,la; INK 0;"*
": BEEP .009,-24: PRINT AT al,la
;" ": IF x=al AND y=la THEN GO T
O 9010
 380 LET mas=INT (RND*150): IF m
as=12 THEN LET pio=1: PRINT AT 1
7,18;"X"
 390 IF pio=1 AND y=18 AND x=17
THEN LET p=p+125: LET pio=0: PRI
NT AT 0,7;p
 400 PRINT AT m,30;"▀"
 410 IF x>m THEN LET m=m+.4: PRI
NT AT m-.4,30;"  ": PRINT AT m,3
0;"▀"
 420 IF x<m THEN LET m=m-.4: PRI
NT AT m+.4,30;"  ": PRINT AT m,3
0;"▀"
 430 LET swe=INT (RND*20): IF m<
x+2 AND m>x-2 AND swe=2 THEN LET
 m=INT m: GO SUB 6000
1000 NEXT v
1001 REM         *******
                 *FINAL*
                 *******
1010 PRINT AT 10,13;"GAME OVER":
 FOR f=0 TO -30 STEP -1: BEEP .0
```

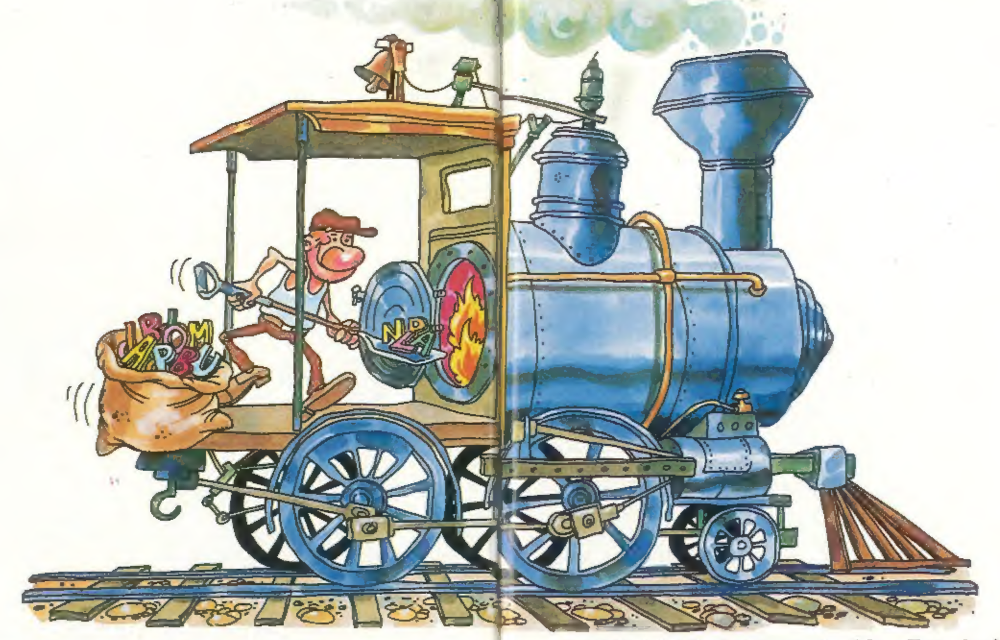


```
5,f: BEEP .05,f+5: NEXT f: PRINT
 AT 21,1;"PARA VOLVER A JUGAR PU
LSA (S)"
1020 IF INKEY$="s" OR INKEY$="S"
 THEN RUN
1030 IF INKEY$="n" THEN STOP
1040 GO TO 1020
5000 CLS : BORDER 3: PAPER 3: CL
S
5010 PRINT AT 0,0; FLASH 1; BRIG
HT 1;"
TRANSON
©MARCO ANTONIO ARROYO RAM
IREZ"
5020 PRINT AT 9,0;"TIENES QUE RE
SCATAR A LA GENTE  QUE CONSIGUE
HUIR DE LA INVASIONDE LAS TROPAS
 RUSAS,Y LLEVARLAS AL BARCO.PERO
 TEN CUIDADO,HAN   PUESTO UN TAN
QUE EN LA MURALLA  PARA EVITAR E
L RESCATE."
5030 PRINT AT 16,2;"       DERECH
A............P           IZQUIE
RDA..........I           PARADO
.............O           ABAJO.
.............A           ARRIBA
.............Q"
5040 PRINT AT 21,1;"PULSA UNA TE
CLA PARA CONTINUAR": PAUSE 0: RE
TURN
6000 LET kkk=(21-ABS m)*8-3
6010 PLOT OVER 1;240,kkk: DRAW O
VER 1;-200,0: BEEP .005,35: PLOT
 OVER 1;240,kkk: DRAW OVER 1;-20
0,0: PRINT AT m,30;" ";AT m-1,3
0;" ";AT m+1,30;" "
6020 IF m+1=x THEN GO TO 9010
6030 RETURN      *******
9000 REM         *FALLO*
                 *******
9010 PRINT AT x,y;"*": FOR f=24
 TO 28: BEEP .005,f: NEXT f
9020 LET h=h-1: LET t=v: IF h=0
THEN GO TO 1005
9030 LET t=v
9100 GO TO 100
```

CARGADOR UNIVERSAL DE CODIGO MAQUINA

J. M. FRAILE

La mayoría de los errores que aparecen en un programa de código máquina se producen, precisamente, a la hora de copiarlo e introducir los datos en el Ordenador. Para evitarlo publicamos este artículo que os servirá de gran ayuda.



A. PERERA

Puede que una magnifica Rutina de código máquina deje de funcionar sólo porque hemos confundido una «O» con un Ø. Para tratar de prevenir este problema hemos desarrollado un completo Cargador de Código Máquina que nos permitirá, a partir de ahora, normalizar la presentación de programas y Rutinas en Código Máquina y minimizar, en la medida de lo posible, la aparición de errores en la introducción de datos.

Estructura y funcionamiento

Todos los programas en código máquina serán presentados con formato Hexadecimal. En aquellos que lo requieran, también será incluido el correspondiente desensable.

Todos los valores hexadecimales que conpongan un determinado programa o rutina, serán agrupados en bloques de veinte cifras, con un número de Líneas y otro de Control. Es lo que denominaremos *Código Fuente.*

Los datos expresados en notación hexadecimal, no tienen de por sí ningún significado para el Spectrum ya que éste es incapaz de trabajar con números que no sean decimales o binarios. Previamente a su utilización, el Código Fuente deberá transformarse en números decimales para que puedan ser entendidos perfectamente por el Ordenador. Esto es lo que llamamos *Código Objeto.*

Esta operación de transformar el Código Fuente (Datos hexadecimales) en Código Objeto se llama «DUMPING» (Volcado en memoria) y la hace automáticamente nuestro programa mediante el comando «Dump».

Una vez tecleado el Programa Cargador, hay que hacer GOTO 9900, con lo que se grabará y verificará en cinta.

El programa se pondrá en funcionamiento automáticamente. Si por cualquier razon, intencionada o no, se detuviese durante su utilización, es imprescindible teclear, «GO TO menú», nunca RUN ni ningún tipo de CLEAR ya que estos dos comandos destruyen las variables y con ellas, el Código Fuente que hubiera almacenado hasta el momento.

Utilización

Una vez cargado desde la cinta, el programa se pondrá en marcha automáticamente, presentando en la línea inferior de la pantalla, un pequeño menú de opciones, a cada una de las cuales se accede pulsando la tecla que corresponde con su inicial.

INPUT. Este comando sirve para introducir nuevas líneas de Código Fuente. Al pulsarlo, el programa nos solicita un número de línea. Obligatoriamente, hemos de comenzar por la línea 1, a no ser que ya hayamos introducido alguna otra previamente.

Tras indicar el número de línea, nos pedirá los Datos correspondientes a la misma. Una vez tecleados (observese un trazo grueso negro que nos ayuda a controlar que el número de caracteres alfanuméricos introducidos sea 20 en todos los casos) y suponiendo que no haya habido ningún error hasta el momento, hay que introducir el Control, que está situado, en cada Línea, a la derecha del Dato. Por último, el programa nos solicita una nueva línea, lo que nos da a entender que todo el proceso anterior ha sido correcto.

En el momento en que se nos solicita nueva Línea o cuando se nos pide el Dato podemos pasar, si lo deseamos, al menú principal pulsando simplemente «ENTER».

TEST. Tiene el doble cometido de listar por pantalla las líneas de Datos que hallamos metido hasta el momento, y de averiguar si una determinada línea ha sido admitida como correcta, ya que si no ha sido aceptada, tampoco aparecerá en el listado.

DUMP. Este comando vuelca el contenido de la variable a$ en memoria, a partir de la Dirección que se especifique. Equivale a convertir el Código Fuente en Código Objeto. Esta operación es obligatoria antes de hacer funcionar una rutina o programa en código máquina. En la mayoría de los casos, con la rutina se indicará también la dirección de memoria donde deber ser volcada y su longitud expresada en bytes. Cuando no se especifique dirección alguna es que la rutina o programa pueden funcionar en cualquier parte de la memoria.

En algún caso puede ocurrir que al intentar volcar el Código Fuente en la memoria, nos aparezca el mensaje «ESPACIO DE TRABAJO». Esto indica que estamos intentando volcar en una zona que el ordenador está usando para sus propios cálculos. Volcar ahí significa la destrucción del programa y todo nuestro trabajo. En tales casos, no queda más remedio que elegir otra dirección de volcado más apropiada.

Durante el tiempo que dura la operación de volcado (depende de la longitud del Código Fuente), se nos muestra en pantalla la dirección inicial y las que restan en ese momento.

SAVE. Este comando nos permite salvar en cinta el Código Fuente (muy importante cuando dejemos el trabajo de tecleo a medias) o el Código Objeto (también llamado Código Máquina) para su posterior utilización. Al pulsar SAVE nos aparecerá un segundo menú de tres opciones: Salvar Código Fuente (F), Salvar Código Objeto (O), indicando Dirección y número de bytes, o volver al menú principal (R). En los dos primeros casos hay que especificar el nombre con el que queremos salvar el Código.

Todas las operaciones de SAVE pueden, opcionalmente, verificarse.

Es muy importante recordar que nunca podrá utilizarse no salvarse el Código Objeto si antes no se ha procedido a su volcado en memoria mediante el comando DUMP.

LOAS. Cuando el número de Datos a teclear sea grande, es normal tener que realizar el trabajo en varias veces. Para ello, puede salvarse en cinta la parte que tengamos (Código Objeto) y luego recuperar mediante la opción LOAD. No es necesario indicar nombre en este caso si no se recuerda. Al cargarse correctamente el Código Fuente, el ordenador nos indicará automáticamente cual fue la última línea que habiamos tecleado y cual es la primera que hemos de introducir ahora.

Los errores

El Cargador de Código Máquina está especialmente estructurado para tratar de prevenir todos los errores típicos de la introducción de datos y que, en el caso concreto de los programas en Código Máquina tienen, por lo general, consecuencias desastrosas, dando al traste con horas e incluso dias de trabajo.

El programa que presentamos chequea las siguientes posibilidades de error:

Que el número de línea no sea correlativo, en cuyo caso se trataría, sin duda, de un error de omisión de línea. Es decir, después de la línea 2, tiene que venir la 3, y no otra.

Que la longitud de la cadena de Datos sea 20. Si es mayor o menor es que sobran o faltan dígitos.

Que las cifras introducidas dentro de una línea de Datos no esten comprendidas dentro del rango de los caracteres utilizados en la notación hexadecimal. Es decir, entre 0 y F. Cualquier anomalía en este sentido será inmediatamente indicada con el parpadeo de la cifra errónea.

Que el control no coincida con la suma de los valores de los Datos en decimal. (Cada dos Datos forman un número hexadecimal).

En todos estos casos, el ordenador nos advierte del error con una señal acústica, a la vez que el borde de la pantalla se vuelve rojo. En situación normal (mientras no se produce ningún error) el borde deberá permanecer siempre blanco.

También hay que tener en cuenta que cualquier error anula la validez de la línea en curso, por lo que habra que repetirla de nuevo correctamente. Para saber las líneas aceptadas en todo momento pulsar Test. A partir de la última, hay que continuar introduciendo nuevas líneas.

Un ejemplo práctico

Como ejemplo de utilización puede valer esta pequeña rutina de Código Máquina que sirve para borrar la pantalla lateralmente. Teclea el Código Fuente. Haz DUMP en la dirección 40000. Salva el Código Objeto desde la dirección 40000, 45 bytes. Haz «BREAK» con «CAPS SHIFT» y «SPACE». Para probar que tanto la Rutina como el Cargador funcionan correctamente, teclea RANDOMIZE USR 40000. La pantalla se verá invadida lateralmente por una cortina azul. Puedes usar esta Rutina en tus propios programas haciendo previamente LOAD " " CODE y llamándola luego desde Basic con RANCOMIZE USR 40000.

```
   2 REM
   3 REM CARGADOR CM MICROHOBBY
   4 REM
   5 CLEAR 65535: LET menu=6000
  10 FOR n=23296 TO 23312
  12 READ c: POKE n,c: NEXT n
  15 DATA 42,75,92,126,254,193,4
0,6,205,184,25,235,24,245,54,65,
201
  70 LET a$="": POKE 23658,8
 100 LET a=10: LET b=11: LET c=1
2: LET d=13: LET e=14: LET f=15
 200 LET li=1: GO TO 6000
1000 REM BUCLE PRINCIPAL
1001 INPUT "LINEA", LINE l$: IF
l$="" THEN GO TO 6000
1002 FOR n=1 TO LEN l$
1003 IF l$(n)<"0" OR l$(n)>"9" T
HEN GO TO 1001
1004 NEXT n: LET line=VAL l$
1005 IF line<>li THEN POKE 23689
,PEEK 23689-1: GO SUB 5000: GO T
O 1000
1007 INPUT "
          DATOS    "; LINE d$
1008 IF d$="" THEN GO TO 6000
1009 LET cx=24-PEEK 23689: PRINT
 AT cx,0;d$;AT cx,21;CHR$ 138;"L
INEA ";li
1010 IF LEN d$<>20 THEN GO SUB 5
000: GO TO 1000
1020 FOR n=1 TO 20
1110 LET w$=d$(n)
1150 IF w$>CHR$ 47 AND w$<CHR$ 5
8 OR w$>CHR$ 64 AND w$<CHR$ 71 T
HEN GO TO 1170
1160 PRINT AT cx,n-1; FLASH 1; O
VER 1;" ": GO SUB 5000: GO TO 10
00
1200 NEXT n: LET ch=0
1210 FOR n=1 TO 20 STEP 2
1215 LET he=VAL d$(n)*16+VAL d$(
n+1): LET ch=ch+he: NEXT n
1250 LET ct=0: INPUT "CONTROL ",
ct
1260 IF ct<>ch THEN GO SUB 5000:
 GO TO 1000
1300 LET a$=a$+d$
2000 LET li=li+1: GO TO 1000
5000 BEEP .2,0: OUT 254,2: POKE
23689,PEEK 23689+1: RETURN
6000 REM MENU PRINCIPAL
6005 PRINT #0; INK 7; PAPER 1;"
   INPUT LOAD SAVE DUMP TEST    "
```

```
6100 LET i$=INKEY$: IF i$="" THE
N GO TO 6100
6200 IF i$="I" THEN GO TO 1000
6210 IF i$="S" THEN GO TO 7000
6220 IF i$="L" THEN GO TO 8000
6225 IF i$="T" THEN GO TO 7500
6230 IF i$="D" THEN GO TO 9000
6250 GO TO 6100
7000 REM SAVE
7001 PRINT #0; PAPER 3; INK 7;"
 FUENTE(F) OBJETO(O) RETURN(R)  "
7002 PAUSE 0: IF INKEY$<>"F" AND
 INKEY$<>"O" AND INKEY$<>"R" THE
N GO TO 7002
7003 IF INKEY$="O" THEN GO TO 72
50
7004 IF INKEY$="R" THEN CLS : GO
 TO 6000
7005 REM SAVE DATA
7006 IF a$="" THEN GO SUB 9500:
GO TO 6000
7008 RANDOMIZE li
7010 LET a$=CHR$ PEEK 23670+CHR$
 PEEK 23671+a$
7015 INPUT "NOMBRE (Save)", LINE
 n$: IF n$="" OR LEN n$>10 THEN
GO TO 7015
7020 SAVE n$ DATA a$()
7025 PRINT #0; PAPER 6;"     DES
EA VERIFICAR (S/N)        ": PAUSE
 0: IF INKEY$="S" THEN PRINT #0;
 INK 7; PAPER 2;" REBOBINE LA CI
NTA Y PULSE PLAY ": VERIFY n$ DA
TA a$(): CLS : PRINT "CODIGO FUE
NTE: ";n$: PAUSE 200
7030 LET a$=a$(3 TO ): CLS
7040 GO TO 6000
7250 REM SAVE CODE
7255 INPUT PAPER 3; INK 7;"DIREC
CION ";di, PAPER 3; INK 7;"N.BYT
ES ";nb
7260 INPUT "NOMBRE (Save)", LINE
 n$: IF n$="" OR LEN n$>10 THEN
GO TO 7260
7270 SAVE n$CODE di,nb
7275 PRINT #0; PAPER 6;"     DES
EA VERIFICAR (S/N)        ": PAUSE
 0: IF INKEY$="S" THEN PRINT #0;
 INK 7; PAPER 2;" REBOBINE LA CI
NTA Y PULSE PLAY ": VERIFY n$COD
E di,nb: CLS : PRINT "CODIGO OBJ
ETO: ";n$'"Inicio: ";di,"Longitu
d: ";nb: PAUSE 200
```

```
7280 CLS
7300 GO TO 6000
7500 REM TEST
7503 IF a$="" THEN GO SUB 9500:
GO TO 6000
7505 CLS : FOR m=1 TO (LEN a$) S
TEP 20
7510 PRINT a$(m TO m+19);" ";CHR
$ 138;"LINEA ";INT (m/20)+1
7520 NEXT m: GO TO 6000
8000 REM LOAD
8010 INPUT "NOMBRE (Load)", LINE
 n$
8020 LOAD n$ DATA a$()
8025 RANDOMIZE USR 23296
8030 LET li=CODE a$(1)+256*CODE
a$(2): LET a$=a$(3 TO )
8035 CLS : PRINT AT 10,5;"Ultima
 linea: ";li-1;AT 11,5;"Comenzar
 por: ";li
8040 GO TO 6000
9000 REM DUMPING
9003 IF a$="" THEN GO SUB 9500:
GO TO 6000
9005 INPUT "DIRECCION ",di: CLS
9006 IF di<PEEK 23653+256*PEEK 2
3654 OR (di+LEN a$/2)>65300 THEN
 PRINT FLASH 1;AT 5,6;"ESPACIO D
E TRABAJO": FOR n=1 TO 200: NEXT
 n: CLS : GO TO 6000
9007 PRINT AT 9,7; FLASH 1;"VOLC
ANDO EN MEMORIA": PRINT AT 7,5;"
Direccion Inicial: ";di
9008 PRINT AT 11,4;"RESTAN ";AT
11,17;"DIRECCIONES"
9010 FOR n=1 TO (LEN a$) STEP 2
9015 POKE di,VAL a$(n)*16+VAL a$
(n+1): LET di=di+1
9018 PRINT AT 11,12;INT (LEN a$/
2-n/2);" "
9020 NEXT n: CLS : PRINT AT 10,8
; FLASH 1;"VOLCADO COMPLETO": FO
R n=1 TO 100: NEXT n: CLS : GO T
O 6000
9500 REM SUBRUTINA MEMORIA VACIA
9501 CLS : PRINT #0; FLASH 1;" N
O EXISTE NINGUN CODIGO FUENTE ":
 PAUSE 300: CLS : RETURN
9900 CLEAR : SAVE "CARGADOR" LIN
E 1: PRINT #0;"REBOBINE LA CINTA
 PARA VERIFICAR": VERIFY "CARGAD
OR": RUN
```

```
1  0E2021005806181120000 246
2  E536121910FBC521D000 1031
3  112000CDB503C1E1E506 1091
4  181120003609191OF8E1 653
5  230D20D9C90000000000 498
9 STOP statement, 67:2
```

CONSULTORIO

Ampliación de memoria

¿Se sabe si la memoria del Spectrum 48K podrá ser ampliable en un futuro? Y si a un Spectrum de 48k se le pusiera la memoria adicional del Spectrum 16K ¿aumentaría su memoria?

José A. CANDIL - Madrid

□ El microprocesador Z-80 no puede direccionar más de 64K de memoria, 16 los ocupa la ROM y los 48 restantes son de RAM; por tanto, si desea ampliar la memoria de su ordenador a más de 48K tendrá que paginarla, además de utilizar RAM estática, ya que el «Refresco de memoria» no funcionaría. Lamentablemente, la solución que usted propone no es viable.

Programar PLOTTER

Me gustaría saber para qué se utiliza el comando DIM. He intentado hacer que funcionara el programa PLOTTER en mi Spectrum y cuando le pongo RUN para que se ejecute, sale «Out of menor y 10 : 1», línea en la que hay dos comandos DIM. ¿Por qué sucede esto?

Luis A. GASCUEÑA - Cuenca

□ Su carta es una de las varias docenas que han llegado a nuestra Redacción haciendo la misma observación y, como suele ocurrir en estos casos, son nuestros lectores los que tienen razón. Por error, el programa PLOTTER apareció como destinado a los modelos de 16K cuando el listado que publicamos sólo funciona en los de 48K. Esto se debe precisamente a la instrucción DIM que reserva una zona de memoria para almacenar los puntos que vamos dibujando. Si quiere adaptar el programa a un Spectrum de 16K bastará con reducir el tamaño de estas instrucciones hasta que sean aceptadas por el ordenador (consulte el capítulo 12 del manual).

Pedimos perdón a todos nuestros lectores a los que haya afectado este error y prometemos vigilar para que no se vuelva a repetir en el futuro.

Llamadas a rutinas

Si tengo una rutina de utilidades, ¿es necesario grabarla aparte?, para que no afecte al programa Basic, ¿dónde la utilizaré?

¿Hay alguna instrucción universal de llamada a las rutinas?

José L. ANDRES - Barcelona

□ Entendemos que se refiere a rutinas en C/M.

Deberá almacenarlas por encima de la RAMTOP tras bajar ésta.

La instrucción general de llamada a rutinas en C/M es la función USR con argumento numérico, que efectúa la rutina colocada a partir de la dirección indicada en el argumento, devolviendo como resultado el valor del par de registros BC en el retorno. El comando a colocar delante de esta función dependerá de lo que quiera hacer con la rutina en particular. Normalmente se utiliza RANDOMIZE, pero en juegos basados en el azar puede dar problemas, a menos que la rutina esté escrita de forma que en el retorno el par de registros BC contengan «cero».

Superampliación

¿Se puede ampliar el Spectrum 48K a 80K poniendo en el bus de ampliación otra de 32K?

Al comprar el Wafadrive y el teclado profesional, ¿necesito comprar también el Interface 1?

¿El Spectrum se puede insertar en el interior del teclado independientemente del Interface 1?

Claudio GARCIA - Sevilla

□ La memoria no sólo hay que ampliarla, sino también direccionarla, por tanto, lo que usted propone es imposible.

El Wafadrive no necesita del Interface 1.

Los teclados comerciales permiten alojar al Spectrum en su interior tanto con el Interface 1 conectado, como sin él.

Problemas de memoria

Al introducir programas de 16K de larga extensión, en sus últimas líneas me salía el mensaje de: E Out of DATA, cuando no me salía: Out of memory. Introduje PRINT 65535-USR 7962 para saber la cantidad de memoria del Spectrum, el resultado fue de: 8704. ¿Esta respuesta es correcta o el ordenador tiene algún fallo?

Jose R. LUKIN - Guipuzcoa

□ Si su ordenador es de 16K, no debe tener problemas al introducir programas escritos para 16K. El informe: E Out of DATA no tiene nada que ver con la capacidad de memoria.

Si quiere saber cuanta RAM tiene su ordenador teclee:

PRINT (PEEK 23732+256*PEEK 23733–16383)/1Ø24

Que le dará el resultado en Kilobytes. Si quiere saber cuantos bytes ocupa un programa , determinado teclee:

PRINT (PEEK 23641+256*PEEK 23642)–(PEEK 23635+256*PEEK 23636)

Adaptación de joystick a nuestros programas

Los programas que publican (concretamente los juegos) ¿son compatibles con joystick, o sólo se puede jugar con las teclas?

Eugenio RUIZ - Madrid

□ Debido a la gran variedad de joysticks disponibles en el mercado, nuestros programas se escriben para ser utilizados con el teclado, pero, normalmente, no le será difícil adaptarlos al joystick que usted posea.

Proteger el teclado

¿Qué se puede hacer para que las letras del teclado del ZX-Spectrum no se borren con el uso?

¿Cómo se puede hacer que el Spectrum utilice lenguajes como el Cobol, etc.?

Luis M. MARQUINA - Madrid

□ Es inevitable que las letras del teclado se borren con el uso. La solución más comúnmente adoptada consiste en cambiar el teclado por alguno de los profesionales que podrá encontrar en las tiendas del ramo.

No tenemos noticias de que existan compiladores de Cobol para el Spectrum, pero podrá usar los lenguajes Pascal, C, Forth, etc., comprando el compilador correspondiente en cualquier distribuidor.

Normas de televisión

Tengo un televisor en color procedente de América y no se por qué mecanismo se ve blanco y negro —algo de un sistema distinto en Europa— mi pregunta es: ¿conectando el Spectrum a ese televisor podrían verse las imágenes en color?

El microprocesador del Spectrum es el Z-8Ø, como el de otros micros, ¿es posible hacer programas equivalentes del Spectrum para esos otros micros?

José Juan RAMOS-La Coruña

□ Efectivamente, en Estados Unidos se usa la norma de televisión llamada NTSC, mientras que en España utilizamos el sistema PAL. Si conecta un Spectrum español a su televisor, lo más posible es que no vea nada.

A pesar de que la mayoría de los micro-ordenadores utilizan el microprocesador Z-8Ø (por ejemplo, el Amstrad y todos los MSX), la compatibilidad del software depende también del sistema operativo, y en el caso del Spectrum, éste es exclusivo.

Mas de 21 UDG

¿Hay alguna manera de hacer más de 21 gráficos definidos por el usuario?

Luis SOTILLOS-Madrid

□ Como ya hemos comentado otras veces, existen varios trucos que permiten trabajar con más de 21 UDGs. Todos ellos requieren un buen conocimiento del ordenador, e implican la alteración de ciertas variables del sistema (CHARS o UDG).

Las amplias explicaciones que requieren se salen del reducido espacio disponible en esta sección, pero son temas que hemos tratado y seguiremos tratando en diversos artículos de la revista.

Televisores antiguos

Rogaría me indicaran como hacer para enchufar el cable de la antena del ordenador a un televisor de los antiguos, que no trae la misma entrada que el cable del ordenador.

José M. ESTEPA-Sevilla

□ Para hacer la conexión deberá utilizar un separador de bandas y adaptador de impedancia que podrá conseguir en cualquier tienda de electrónica. La firma TELEVS fabrica un modelo bajo la referencia 51.616. Este aparato incluye las clavijas de antena a 75 ohmios y un trozo de cable coaxial para conectar el cable del ordenador al adaptador.

Lo mejor es que acuda a la tienda de electrónica y lleve el cable del ordenador.

Otra posibilidad es que un técnico le instale en su televisor una toma de antena a 75 ohmios.

Ampliación de memoria

Quisiéramos saber dónde podríamos ampliar el Spectrum de 16 a 48K, a ser posible, en la provincia de Valencia, y cuanto cuesta aproximadamente.

Andrés y Juan-Valencia

□ Les recomendamos que busquen en la publicidad de nuestra revista el concesionario de HISSA más cercano a su domicilio. Ellos mismos les informarán del precio de la ampliación.

DE OCASION

● VENDO por comprarme un Flopy, Interface 1, más microdrive, tres cartuchos, nuevo con caja e instrucciones por 30.000 ptas. También vendo VIC 20 nuevo con caja e instrucciones esp. 3k5 RAM por 15.000 ptas. José Jover. Tlf. (972) 231742 (GERONA).

● VENDO ZX81, ampliación 16K, inversor de vídeo, tecla de repetición, cintas de Investrónica de Indescomp, tres libros sobre su uso y su manejo y uno y uno del CM, también regalo al comprador una película en vídeo sistema VHS «Ratas del asfalto» original (no copiada de la tele) valida para club de vídeo. Acepto o cambio por Spectrum 16K y pagaría 4.000 ptas. más. Precio del lote 20.000 ptas. Enrique García Batalla. Avda. de Eduardo Castro 149 Gijón (ASTURIAS).

● VENDO ZX Spectrum 48K con teclado progesional, en 50.000 ptas. También vendo ZX Spectrum 16K con memoria externa para 48K, por 35.000. Para contactar llamar al tlf. (956) 401708, o escribir a Manuel Rodriguez Santos. Bd. Torno blq. 14. 2° A Chiclana (CADIZ).

● COMPRO las instrucciones del «Hobbit», no importa el idioma en que esten. Eduardo. Apartado 920 La Coruña.

● DESEO ponerme en contacto con usuarios del ZX Spectrum y ZX 81, de la comarca de «Rivera de Duero». Tengo especial interes en programas de utilidades o Work-Games. Dispongo de más de 100 juegos. Me interesa por el código máquina. Interesados ponerse en contacto con Juan Carlos Cilleruelo Gonzalo. Capitán Rojas 12. Peñafiel (VALLADOLID). Tlf. (983) 880811 (fines de semana).

● ME GUSTARIA recibir información sobre programación: Código Máquina y de los posibles lectores y nociones de Basic. También deseo recibir programas y pequeños trucos de los lectores. Escribir a Juan C. Triana. Sinieso Delgado, 5, 2° D. 34004 Palencia.

● CAMBIO mesa de mezclas de 12 canales, más 2 auxiliares marca Dinamix 212 nueva, tres meses de uso, valorada en 89.000 ptas. por Spectrum Plus más impresora o Spectrum 48K, impresora y cassette. Acepto otras ofertas. Contactar con Juan Carlos de Cabanyes Ortega. Apartado 122. 34080 Palencia. Tlf. (988) 808243.

● VENDO videojuego Atari 2600 con tres juegos, cuatro mandos, el transformador, un catálogo e información sobre precios. Todo por 22.000 ptas. O cambio por inteface 1 e interface 2. O bien interface 2, dos mandos del Joystick y cinco cintas de juegos o utilidades. Interesados escribir a Alberto Cardona Cabrera. Avda. del País Valenciá, 207. Benissa (ALICANTE).